January 2022

Segmentation Of Intracranial Structures From Noncontrast Ct Images With Deep Learning

Evan Porter
Wayne State University

SEGMENTATION OF INTRACRANIAL STRUCTURES FROM NONCONTRAST CT IMAGES WITH DEEP LEARNING

by

EVAN PORTER

DISSERTATION

Submitted to the Graduate School

of Wayne State University,

Detroit, Michigan

in partial fulfillment of the requirements

for the degree of

DOCTOR OF PHILOSOPHY

2022

MAJOR: MEDICAL PHYSICS

Approved By:

____

Advisor Date

____

____

____

____

DEDICATION

For John Rodger Porter Jr.
1954-2009

ACKNOWLEDGMENTS

- **Anne Schutter**, for fostering my scientific curiosity and supporting me throughout my graduate experience. I hope I can measure up to the level of medical professionalism you showed throughout your career.
- **Thomas Guerrero**, without your mentorship none of this would’ve been possible.
- **Kathy Guerrero**, who kept my stomach and heart warm.
- **Zaid Siddiqui**, your mentorship, collaboration, and feedback were instrumental in shaping my skills to what they are today.
- **Nick Myziuk,** because Nick insisted, I recognize his endless disturbances while I prepared this manuscript.
- **Derek Mumaw, Andrea Bisutti, Ron Levitin, Drew Thompson, Carl Stanhope**, for the joy you’ve brought to our otherwise gloomy basement.
- **Geri Blake** of the TCIA, for her tireless curation of over 560,000 individual DICOM files uploaded in our data set.

TABLE OF CONTENTS

LIST OF TABLES

Appendix Tables

LIST OF FIGURES

Appendix Figures

TABLE OF SYMBOLS

Symbol	Description
ρ	Physical density (g/cc)
Gy	Gray (J/kg), a unit of absorbed dose
$\sum_{i}^{n} x$	Summation of x over $[i, n]$
$\cap$	Binary operator for intersection (logical AND)
ε	Represents a small value, $\varepsilon \to 0$
~	Binary operator for negation (logical NOT)
sup	Supremum of a set
inf	Infimum of a set
$\in$	Denotes set membership
$\lvert x \rvert$	Cardinality of a set
α	Asymmetry scalar
β	Magnitude of asymmetry bias
Δ	Difference or loss function
λ	Learning rate

CHAPTER 1 The Evolution of Artificial Intelligence

1.1 Ambitions Through History

In his seminal work *The Structure of Scientific Revolutions*, Kuhn challenged the narrative that scientific advancement occurs through a gradual and continual process. Instead, he described the advancement of scientific thinking as long periods of the status quo, or “normal science”, where the focus is on accumulation of knowledge for an existing scientific theory and the ever-increasing complexity in puzzle solving. Then, the periods of normal science are interrupted by landmark developments, referred to as paradigm shifts, which spur rapid progress. These shifts alter how problems are solved, the complexity of solutions, how existing data is analyzed and the future roadmap of investigation in a field. Often, the catalyst for a paradigm shift is advances in logic, new availability of data, advances in adjacent areas of study or improvements in computational technology. Historical examples of paradigm shifts include the transition from classical to relativistic mechanics used to explain high velocity systems or the shift from Ptolemy’s to Copernicus’ model of the solar system, vastly simplifying planetary dynamics.

In many fields of research, artificial intelligence has unlocked new approaches to data analysis and problem solving, making it the zeitgeist for the last half decade. While artificial intelligence has recently pervaded contemporary discourse, the aspirations and applications for artificially intelligent systems have existed for millennia. Only recently has the combination of generalizable computational frameworks and affordable, powerful hardware allowed for the widespread adoption of artificial intelligence, thus sparking the paradigm shift many fields are currently experiencing.

Ancient writers and philosophers have long imagined the potential for artificially intelligent systems to assist humans, augment our abilities or perform superhuman tasks. But as is a common theme throughout history, visionaries are often constrained by the limits of technology. In *The Illiad*, Homer described the workshop of Hephaestus, the god of blacksmithing, who created golden tripods, called automatons, which could be programmed by wrapping spools of rope around each wheel. With careful programming, the spools would unwind, and these tripods would autonomously enter Mount Olympus, serve the gods, and return to the workshop again. But Hephaestus greatest automaton was Talos, a giant crafted from bronze, built to protect Europa on the island of Crete. Talos would thrice daily circumnavigate the island and throw boulders at pirates and marauders who attempted to kidnap Europa. In one rendition of the story, Medea tricked Talos into believing he could become immortal if Medea removed the nail from Talos's ankle. Unbeknownst to Talos, within him ran a tube from head to toe containing *ichor*, the ethereal blood of the Gods, and the nail plugged the only opening. So, when Medea deceived Talos and removed the nail, the life-giving fluid within Talos drained and he became but bronze once again.

Three hundred years later, Aristotle mused that instead of assisting humans, Hephaestus's automated tripods could be used to replace humans in the most mundane of tasks, freeing slaves from tending fields and cleaning homes:

> "There is only one condition in which we can imagine managers not needing subordinates, and masters not needing slaves. This condition would be that each instrument could do its own work, at the word of command or by intelligent anticipation, like the statues of Daedalus or the tripods made by Hephaestus, of which Homer relates that "Of their own motion they entered the conclave of Gods on Olympus", as if a shuttle should weave of itself, and a plectrum should do its own harp playing." - Aristotle

Although an ambitious vision, Aristotle conceded that the technology simply did not exist to automate labor and his idea of automated labor freeing the slaves remained but a dream. So, as would be the case throughout much of history, the paradigm did not shift and throughout the next millennium, artificial intelligence would remain but a fantasy for writers and philosophers.

In the 16th century, mechanical parlor tricks drew attention to entertain and mystify royalty. Experts of the time designed intricate and unique machines to imitate the external behavior of intelligent life. One of the most spectacular examples was Jacques de Vaucanson's Digesting Duck (Figure 1.1), which was able to quack, flap its wings, digest grain, and move its head. To Homer and Aristotle, this duck would likely have been artificially intelligent. But, to our contemporary understanding, we would not categorize Vaucanson's duck as artificially intelligent, just as we don't apply that designation to other complex mechanical devices like typewriters or clocks.

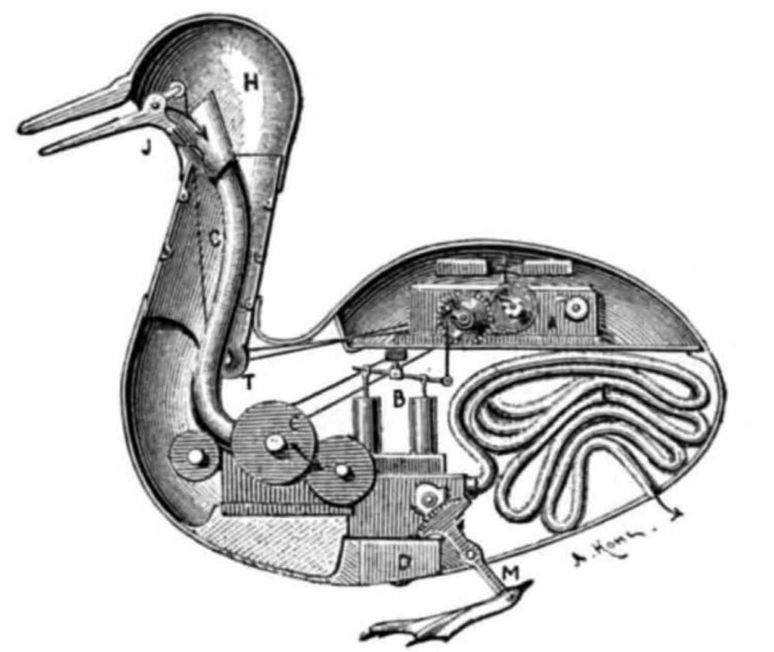


Figure 1.1: A lithograph of Vaucanson's automatic Digesting Duck. Inscriptions designate clockwork (A), pump (B), mill for grinding grain (C), intestinal tube (F), bill (J), head (H) and feet (M). (Public Domain).

Furthermore, while the Digesting Duck displayed an increasing mastery of mechanical design, this approach was decidedly non-generalizable. The creation of the Digesting Duck required immense effort to design and create the unique solution to the task of automating a duck. For artificially intelligent systems to become widespread, a more generalizable approach to encoding and solving problems was first required.

At the same time as Vaucanson was mystifying aristocrats, the German mathematician Gottfried Wilhelm Leibniz published a dissertation titled *On the Combinatorial Art*. In this work, Leibniz proposed a generalizable, logical symbolic framework capable of solving any task. Born out of the idea that all human thought was comprised of logical subcomponents (akin to letters within a word), Leibniz proposed deconstructing any idea into its constituent parts. This logical framework would allow for the user to encode their thoughts, and thus their problems, into a system of fundamental logical variables. The mathematical field of symbolic combinatorics could then be applied to this system of variables to solve for the solution to the system of thought equations. Finally, the resultant solution could be re-encoded back into a human level thought to answer the given problem. Leibniz envisioned that this machine could be used to solve all intellectual problems and when debate arose, one could proclaim "lets calculate", encode their problem into the machine and compute the definitive answer.

In essence, Leibniz had theorized a primitive programming language, but like his predecessors, Leibniz was constrained by the state of his era's computational power. In fact, throughout his life Leibniz became more disillusioned with the idea of a general, logical calculator. Ultimately, the closest he got to achieving his vision was the 'stepped reckoner', a simple mechanical calculator with a decimal registry capable of 8-digit addition, subtraction, multiplication, and division. For the next two-hundred and fifty years, mechanical computers grew in complexity, but like Vaucanson's duck and Leibniz's stepped registry, they required complex and unique design to solve individual tasks with predetermined solutions. What was required was a more generalizable framework of encoding and solving problems, forgoing the need to devise a unique solution to every new problem. To facilitate this computational framework, computing devices orders of magnitude more powerful than the stepped reckoner would be required.

1.2 Scientific Inspiration

The investigations into modern artificial intelligence were in the late 1940s. These researchers, mostly mathematicians by training, were inspired by advancements spilling over from other scientific fields, namely neuroanatomy (e.g., neural connections), psychology (e.g., behavioral theory), and electrical engineering (e.g., vacuum tubes, magnetic tape drives, ferrite core memory).

In 1906, the Nobel Prize in Medicine was given to two Spanish neurologists, Santiago Ramón y Cajal and Camillo Golgi, in recognition of their work on the structure of the nervous system. Cajal share of the award was for his discovery that each neuron within the nervous system behaved as a unique entity, and complex actions of the system resulted from synapses sharing impulses throughout the system. Each neuron is comprised of a body, dendrites and long tail(s), the axons, which nearly connect to the dendrites of the surrounding neurons. Throughout the nervous system, every neuron is constantly sending or receiving electrical pulses through its dendrites or axons. Depending on the frequency and strength of these pulses, the neuron may either activate, continuing the pulse onto its neighbors, or not. It is only through a system of billions of these simple, binary actors does the complexity and intelligence of evolved life begin to emerge. In response to this discovery, Canadian neurologist Donald O. Hebb proposed that when a dendrite-axon pair is frequently simulated, it induces changes within the cells and increases the expression of this synapse. His proposal went on to claim that the repeated activation of neurons changed their expression to stimuli and their weighting within the system, thus resulting in learning.

The popularity of behavioral theory and psychology grew in the early 20th-century, and while many of the scientific "developments" of the field were motivated by biases and prejudices, the work of B.F. Skinner attempted to legitimize the scientific method within psychology. Skinner focused his career on the study of human free will and what he called “reinforcement theory”. In essence, Skinner theorized that human free will was illusionary and instead an individual’s current personality was a function of past reinforcement of specific behaviors. He derived this theory from the lens of evolutionary biology, where certain traits are expressed in a species through subsequent generations of survival (reward) or death (punishment). In Skinner’s theory, reinforcement could come as acute or continuous reinforcement and could be either negative (punishment), positive (reward) or extinction (absence of rewarding) which also weakened behavior. Therefore, Skinner claimed that with proper incentivization in place, an individual could be conditioned to strengthen a desired behavior.

John Ambrose Fleming, an electrical engineer, worked at a transatlantic radio company in 1904. When tasked with improving signal strength, Fleming looked back at his experiences at the Edison Electric Light Company and devised a variant of the electric lightbulb where the heated electrical filament generated thermionic emissions, and the electrons were attracted to a positive plate (the anode). Due to the difference in charges, the flow of current was restricted to one direction and the current could be quickly switched on or off by changing the relative potential voltage between the filament (cathode) and the anode. Fleming’s invention, the vacuum tube, was successful in improving the signal-to-noise ratio of the transatlantic radio, and would be eventually used in nearly every advanced electronics system of the era. This simple lightbulb derivative would become foundational to the early development of electronics and computer engineering.

While neuroanatomy, psychology and electrical engineering drew little inspiration from one another at the time, at the intersection of these fields was the bourgeoning study of computer science and artificial intelligence. These researchers possessed the same aspirations of the many dreamers who came before them, but due to the surrounding new technology, they found themselves at the precipice of a paradigm shift that would revolutionize how problems were solved.

1.3 Early Investigations into Artificially Intelligent Systems

Early pioneers in the artificial intelligence sought to harness computational power to model human intelligence, but this goal was reliant on first defining human intelligence. A reasonable starting point was to determine tasks, games or activities which were expressive of key traits of intelligent life. Examples of the toy problems, intelligent traits they aimed to express, and researchers involved in that field are given in Table 1.1.

Table 1.1: Original problem classes in artificial intelligence research and the pioneers who advanced the fields.

PROBLEM CLASS	EXAMPLES	RESEARCHERS
PLAYING GAMES	Chess and Checkers	Allen Newell, Arthur Samuel
PATTERN RECOGNITION	Numeral identification, Shape Identification	Gerald Dinneen, Oliver Selfridge
NATURAL LANGUAGE PROCESSING	Language translation, Artificial language creation	John McCarthy
NEUROLOGICAL MODELING	Neural networks to simulate cognition	Karl Lashley
SYMBOLIC PROCESSING	Writing symbolic logic proofs	Allen Newell, Herbert Simon

From our contemporary view, a few of these toy problems may appear to be quite simple (e.g., playing checkers), but it is important to fully consider the state of computers in the 1950s. From 1954 to 1963, the world’s most power computer was the Naval Ordnance Research Computer (NORC), which was owned and operated by the US Navy Bureau of Ordnance. This supercomputer could complete 15,000 operations a second (compared to trillions, or more, now), it possessed 3600 words of memory (64 bits per word) and cost $2.5 million (in 1950s money). Recognizing the limitations of computers from the start, in the official proceedings the first conference of artificial intelligence held in Los Angeles in 1955, the chairman wrote:

> “This group of papers suggests directions of improvement for future machine builders whose intent is to utilize digital computing machinery for this particular model technique. Speed of operation must be increased manyfold;

> simultaneous operation in many parallel modes is strongly indicated; the size of random-access storage must jump several orders of magnitude […] With such advancements and techniques discussed in these papers, there is considerable promise that systems can be built […] which will imitate considerable portions of the activity of the brain and nervous system." – Willis Ware [1]

Therefore, the work conducted by researchers of the time was limited to even more primitive computers, custom built systems or what limited computational power they had available. This unfortunately meant that many problems remained computational infeasible at the time and researchers had to resort to other means of testing their hypothesis. In a particularly dramatic example, Herbert Simon, an early pioneer in symbolic processing, prototyped his first thinking machine by enlisting his children to simulate the working register of a computer.

An early pioneer of pattern recognition was Oliver Selfridge who devised a hierarchical approach to pattern recognition. At the foundation of an artificial intelligence algorithm were "data demons" who were responsible for the basic digestion of incoming data: edge enhancement, vertical line, horizontal line, or vertex recognition and so forth. The post-digested data was then passed upwards to subsequently higher ordered demons who identify higher order features, like squares or triangles. Selfridge described each middle demon's identification of a feature as a shout, where the loudness of their shout was proportional to their confidence in having identified that feature. In the second highest order of this pyramidal system existed the cognitive demons who convert these series of higher order features into complex concepts, such assembling a collection of shaped related shouts into a number. Finally, the highest order demon would synthesize the information from the lower order demons and make the final decision for the classification of the image which was originally provided.

Selfridge proposed that this collection of demons could be incrementally improved by amplifying the shouts of certain demons over that of others. Alternatively, modifications could be made by replacing the higher-order demons, allowing for new interpretation and classification of the digested data. Furthermore, behavior could be reinforced by replacing the demons which overall shout the least and are thus least useful. These demons could then be mutated, or fully replaced, with the hopes of evolving to a new demon descendent who proves more useful. Ultimately Selfridge realized two notable limitations to his proposed demon model: design and computation. For Selfridge’s model to work, it required designing individual demons to impose unique abilities upon them, as he did not have a way for these demons to learn their behaviors artificially. Additionally, the ambitions of Selfridge’s model far exceeded the computational power of the general computing systems of the time. To overcome this, Selfridge attempted to build an electrical prototype of his model where inputs of binary images were encoded as a series of connected wires in a grid (like a switch board) with the final output displayed by a series of lights. The demons were then subsystems of vacuum tubes which performed designed computational tasks, such as line or vertex identification. Another shortcoming of Selfridge’s design was that it heavily relied on human intuition and perception to create the individual demons that together worked proficiently as a system. Alternatively, the idea of mutation and replacement required immense, and unobtainable, computational power. The accumulation of these limitations and workarounds resulted in Selfridge’s demon collection becoming akin to the Vaucanson’s digesting duck – a manually and uniquely designed system capable of mimicking the behavior of a human but incapable of generalized learning. Despite lacking generalizability, he had imagined, Selfridge’s preliminary theorization for an artificial intelligence system for use in pattern recognition was revolutionary. In fact, as we will discuss later, our contemporary understanding of deep neural networks has not strayed far from Selfridge’s proposal more than 65 years ago.

As Selfridge and the other researchers of the 1950’s and 1960’s ran up against the computational ceiling, they began to realize their bold ambitions for artificial intelligence were unobtainable at the time. Many researchers shifted their focus to other tasks and advancement stalled. Artificial intelligence had again fallen into a period of reduced academic focus and development.

1.4 Convolutional Neural Networks

For over thirty years, Selfridge's numeral recognition problem remained mostly unobtainable until two algorithmic discoveries solved the problems of generalization and computational requirements. The first key discovery was an algorithmic change proposed to entirely invert the existing paradigm of training an artificial intelligence system. Instead of evaluating the least active node within the neural network from the bottom up, the error in a prediction can instead be back-propagated top down through the model [2]. Using the same incentivization philosophy proposed by B.F. Skinner, the training of a neural network would be dictated by a loss function, which would compute the error between a model's prediction and the known ground truth. Then, the partial gradient for each weight within the model would be computed and the contribution of each model weight to the error would be determined. With the known contributors to the prediction's error, these weights could then be altered in relation to their contribution, thereby making future predictions less likely to repeat that error. While this concept seems conceptually simple (the paper was only three pages), the impact it had on the computational efficiency of training neural networks was substantial. Prior to backpropagation, the loss function was a system of equations relating to each individual weight in the model. To compute the changes to every weight in a 1,000-parameter model, a separate prediction would be required for each loss-weight pair. Therefore, the data would need to pass through the model 1,000 times to modify each of the weights individually. Contrast this with the backpropagation paradigm where the data only needs to pass through the model a single time and then the individual model weight gradients of the loss function are backpropagated throughout the network. This discovery completely changed the computational complexity of artificial intelligence research and overnight allowed for far more complicated neural network designs.

Soon after the discovery of backpropagation, more complex and sophisticated neural network designs began reviewing unsolved problems of the past. One such problem was that of numerical recognition, such as handwritten zip codes on postcards and magnetic numerals on bank checks. Spurred on by the discovery of backpropagation and the newly unlocked complexity of trainable models, researchers began to apply more complex mathematical operators to neural network designs. One such addition was the convolutional operation to create a “convolutional neural network” [3]. Instead of a system constructed with multiplication and addition operators, the convolutional neural network allowed for the training of the convolutional kernel. These convolutional kernels could then learn pattern, texture or edge detection as needed to satisfy the loss function. Unlike Selfridge’s model where the data digesting demons were manually, individually designed to detect edges or vertices, trainable convolutional operations were able to artificially learn those features as a cohesive system. The impacts of convolutional neural networks were immediately apparent, with the first application able to correctly identify 16x16 pixel handwritten digits with only a 1% error rate on the test set. Within years, these convolutional neural networks quickly revolutionized the United States Postal Service and the check processing industries. Despite these nearly immediate impacts in banking and postal service, the 16x16 pixel images were too small to find useful applications in radiation oncology or medical image analysis.

1.5 The Contemporary Paradigm

What was required to make artificial intelligence widespread was the availability of inexpensive and powerful computational hardware designed to calculate convolutional operations, which are simply matrix multiplications. It just so happens that in the late 1990’s home desktops were becoming a popular platform for at-home video games and the demand was increasing for high-quality graphics. To facilitate this, silicon chip manufacturers, like Advanced Micro Devices (AMD) and Nvidia Corporation, began designing specialized computational devices for computer graphics. Nvidia’s first offering, the GeForce 256 was released December 13, 1999, and AMD released The Radeon on April 1, 2000.

In essence, the graphics displayed in a video game are comprised of numerous triangles, which the vertices are stored as matrices. When players input changes to the video game, the rotations and translations are applied to the vertex matrix through matrix multiplication. To achieve a responsive video gaming experience, the matrix multiplications necessary to update the screen needed to be completed in a fraction of a second. The strength of a computer’s CPU is the ability to conduct many tasks within the computer, but this makes it poorly suited to repeat a simple task many times, such as matrix multiplication. Therefore, manufacturers designed GPUs with relatively primitive computational cores, but were able to fit 100’s or 1,000’s of these nodes in parallel. This meant that the specialized hardware in GPUs could compute matrix multiplications orders of magnitude quicker than CPUs. The only issue was GPU manufacturers did not make the graphics drivers, the software which communicates with the hardware, available to researchers to harness the GPU’s power for other tasks.

In 2007, Nvidia released Compute Unified Device Architecture (CUDA), an application programming interface (API) designed to allow for the leveraging GPU hardware for general computing applications. The release of CUDA unlocked the power of GPU hardware for researchers and removed the last computational barrier of widespread machine learning adoption. Suddenly, cards costing hundreds of dollars were able to compute matrix operations with similar performance to CPU bound supercomputers. This allowed artificial intelligence research to push the envelope even further with model complexity and size, allowing for even more complex tasks to be solved. Building off the CUDA API, further tools (e.g., PyTorch, TensorFlow) were designed with more user-friendly interfaces and implementations of recently published scientific developments. Thus, these libraries democratized the power of the GPU, allowing non-computer science researchers to explore domain specific artificial intelligence and ignited a paradigm shift across many fields.

CHAPTER 2 Deep Learning Fundamentals

2.1 How Does a Machine Learn?

Traditional problem-solving algorithms define a problem and a specific set of steps required to arrive at a solution [A]. In contrast, a deep learning model is a statistical framework, which when trained, stochastically arrives at a solution. For the model to effectively converge to a solution, it must be able to evaluate the quality of candidate solutions as it learns. Loss functions, also called objective functions or cost functions, quantify the quality of a candidate solution during the model training process. For each step during training, the model's weights are progressively updated to yield predictions which minimize the loss function. Because the loss function dictates the model's measure of success and the degree to which the weights are updated, choosing the proper loss function for a given task is vital.

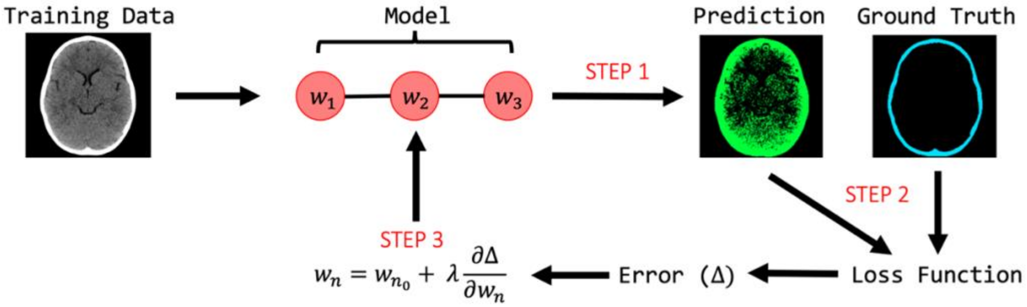


Figure 2.1: The steps in training a deep learning model. Step 1, from the training data, a prediction is made. Step 2, using the loss function, the ground truth and prediction are compared, and an error is determined. Step 3, each weight is updated proportionally to the gradient of the error.

At the beginning of training a model, the weights are randomly initialized and generally incapable of making any useful predictions. However, through backpropagation training, models can learn to solve tasks across many divergent domains. Take, for example, the simple problem of segmenting the skull on a CT image, as shown in Figure 2.1.

[A] Contents of this chapter were previously published in: Porter E, *et al.* Effect of Loss Functions in Deep Learning-Based Segmentation. *Auto-Segmentation for Radiation Oncology.* CRC Press. 2021; 133-150.

The backpropagation training process is broken into three steps: prediction, evaluation and back-propagation. During the first step, the training input data flows through the model which is simply a series of mathematical operations, most commonly convolutional operations. The data which returns from the model is referred to as a prediction. In the skull segmentation example, we provide the model with a two-dimensional CT image slice as input, from which the model generates a prediction for a segmentation mask. From the example in Figure 2.1, we can see that the current model's skull prediction is non-ideal and further training, or updates to the model's weights, is warranted. Next, the error of the prediction, in relation to the ground truth, is calculated using the loss function. In the final training step, the gradient of the error is calcualted with respect to each model weight. Then every weight is updated by the scaled gradient of the error, with the intent of minimizing each weight's contribution to the error in subsequent predictions. The scaling factor, commonly called the learning rate, is represented by λ in Figure 2.1. Therefore, to allow for backpropagation training, a loss function must have scalar-valued output and be differentiable with respect to the model weights. A complete training process repeats these three steps until the output of the loss function, or prediction error, is minimized. Ideally, upon finishing training, the model weights should converge upon a state capable of robustly solving the given task.

In addition to dictating what is learned, a loss function can influence how easily a model converges upon a solution. Like many optimization problems, the training of deep learning models utilizes a multi-dimensional gradient descent. A simple visual representation of the training process would be the act of navigating to the lowest point on an uneven plane, such as those shown in Figure 2.2. If the plane possesses many depressions in addition to the true lowest point, it would be difficult to know if we are at the lowest point globally or merely locally; after all, our only knowledge is of our local surroundings, not if there is a deeper depression elsewhere on the plane.

To adapt this to deep learning terminology, the x-y axis of the surface represents all potential model weight combinations, and the z-axis indicates the loss function performance of the current weight combination. During training, the model is initialized randomly within the weight possibility space. Then, as the model trains, it explores the space of its possible weight combinations to minimize the loss function. Optimal loss functions therefore have an easily computed gradient path towards the global minimum.

We refer to the set of weights which minimize the loss function as the global minimum, and the other sets of weights which produce loss functions lower than their surroundings as the local minima. If we chose a loss function completely unsuited to the data, it is unlikely the model will train at all, with a visualized loss space [4] example given in Figure 2.2.A. If we instead chose a poorly suited, but trainable, loss function, there will be both a global minimum and local minima, as in Figure 2.2.B. But, if we carefully choose a loss function well suited for our task, finding the global minimum will be both simple and efficient, as seen in Figure 2.2.C.

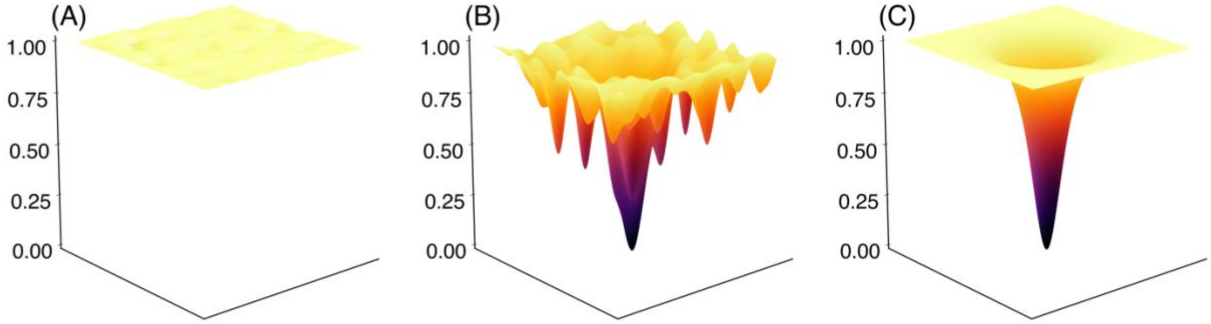


Figure 2.2: A visual depiction of loss functions where the x-y axis is model weight combinations, and the z-axis is the loss function. With an incorrectly chosen loss function (A), a poorly suited loss function (B) and an easily trainable loss function (C).

A well-chosen loss function has a significant role in reaching an optimal solution for a given deep learning task. In this chapter, we will cover: the necessary elements of a loss function, presenting a segmentation task for a loss function, common loss functions and their applications, dealing with imperfect data, choosing a starting loss function, and troubleshooting methods to help overcome frequent challenges in medical image segmentation.

2.2 Admissibility of a Loss Function

To understand the importance of admissibility, let us imagine that two people are bidding to build a fence enclosure for a farmer's sheep. The farmer only tells both designers that whoever designs the fence with the shortest length will be hired. The first designer, using his knowledge of geometry, designs a circular fence, large enough to encircle the flock. On the other hand, the second designer proposes to build a fence only around himself, declaring himself 'outside' the fence. Clearly, this second solution fails to enclose the flock, which is the original purpose of a building fence. However, the farmer presented the ideal solution as that which minimized fence distance, not that which minimized the danger to the sheep. In a deep learning context, the farmer's loss function, length of fence, was not admissible to his true intentions behind building the fence.

While the second solution may seem outlandish, deep learning models are inherently prone to converging upon these 'lazy' solutions. For segmentation tasks, common 'lazy' solutions are models which do not predict every structure, predict highly smoothed structures or models which uniformly predict a single structure. To prevent theses 'lazy' solutions, we must carefully choose a loss function which defines our ideal solution to the task, minimizes the risk of unintended results and ensures effective convergence to a robust solution.

2.3 Presenting the Problem

The remainder of this chapter covers the proper combination of ground truth data and loss functions and presents a selection of different losses useful for image segmentation. For our discussion, we consider a segmentation ground truth to be a label mask where each voxel is designated as either a member of the class or not. These ground truth label masks can be organized as either a multi-label or multi-class segmentation tasks, both of which can be used to train a deep learning model.

A multi-label segmentation allows for each voxel to be a member of multiple classes, as well as not a member of any class. An example of a multi-label segmentation is a patient with of multiple thoracic structures and a body contour. In this case, every voxel classified as ‘heart’ would also be member to the ‘body’ class. And, for any voxel exterior to the body, class membership would not be required.

A multi-class segmentation is a restriction of a multi-label segmentation task, where each voxel is a mutually exclusive classification. This means that each voxel must, and can only, be a member of a single segmentation class. For example, if you are contouring the left and right lung, each voxel will be one of three classes: left lung, right lung or neither lung. Through the inclusion of the ‘neither’, also referred to as the ‘background’ class, the problem allows for every voxel to be a member of a class. To restrict voxels from having membership to multiple classes, or likewise to reduce a multi-label to a multi-class segmentation problem, binary operators (i.e. AND, OR, and NOT) can be utilized.

Strict adherence to the multi-class labeling rules is important because any mislabeled voxels will interfere with the model’s training. Take, for example, a voxel which was not assigned any of left lung, right lung or neither. During the training process, a prediction of any class membership will falsely be evaluated as an error and will be backpropagated into the model weights, potentially interfering with the otherwise properly trained parameters.

Although multi-class labeling restricts the preparation and data organization of the ground truth labels, doing so also restricts the complexity of any prediction. By reducing the degrees of freedom possible in a solution, the overall solution space is restricted and the gradient decent is simplified. This means that, for most tasks, preparing the ground truth as a multi-class problem will result in quicker convergence to a solution.

As depiction of both label types, Figure 2.3 demonstrates different representations of an arbitrary 2D image composed of a partially overlapping circle and triangle. Figure 2.3.B shows a "one-hot encoded" multi-label data set representation of the original image, Figure 2.3.A. In this case, a third dimension is added to the 2D image, with each position along this dimension called a channel, where each channel represents membership of the pixel position to different categories, or classes, of data. A pixel value of 1 in channel 1, Figure 2.3.B left, would indicate that the pixel belongs to the circle region, and a pixel value of 1 in channel 2 would indicate that the pixel belongs to a triangle region. It is important to note that in a multi-label representation of the data, a given pixel position may hold a value of 1 in either channel, indicating that the pixel position belongs to both the circle region and triangle region. This contrasts with multi-class representations of the data set, which must hold mutually exclusive classifications. In Figure 2.3.C, a multi-class label-encoded data representation of Figure 2.3.A is shown. In this representation, a unique integer label is assigned to each pixel, which indicates which classification category the pixel belongs to: 0 – background, 1 – circle only, 2 – triangle only, 3 – intersection region of the circle and triangle. Because this is a multi-class representation, a new classification is needed to indicate membership of the pixel in the overlapping region. In Figure 2.3.D, a one-hot encoded multi-class representation of Figure 2.3.A is shown. In a similar fashion to Figure 2.3.B, multiple channels are again utilized to indicate the category a given pixel belongs to (from left to right): channel 1 – background, channel 2 – circle only, channel 3 – triangle only, channel 4 – circle and triangle intersection. As will be discussed later, though similar in their composition, the use of either a multi-label or multi-class representation (Figure 2.3.B vs. Figure 2.3.D) for one’s data set may hold distinct advantages for loss functions and their application.

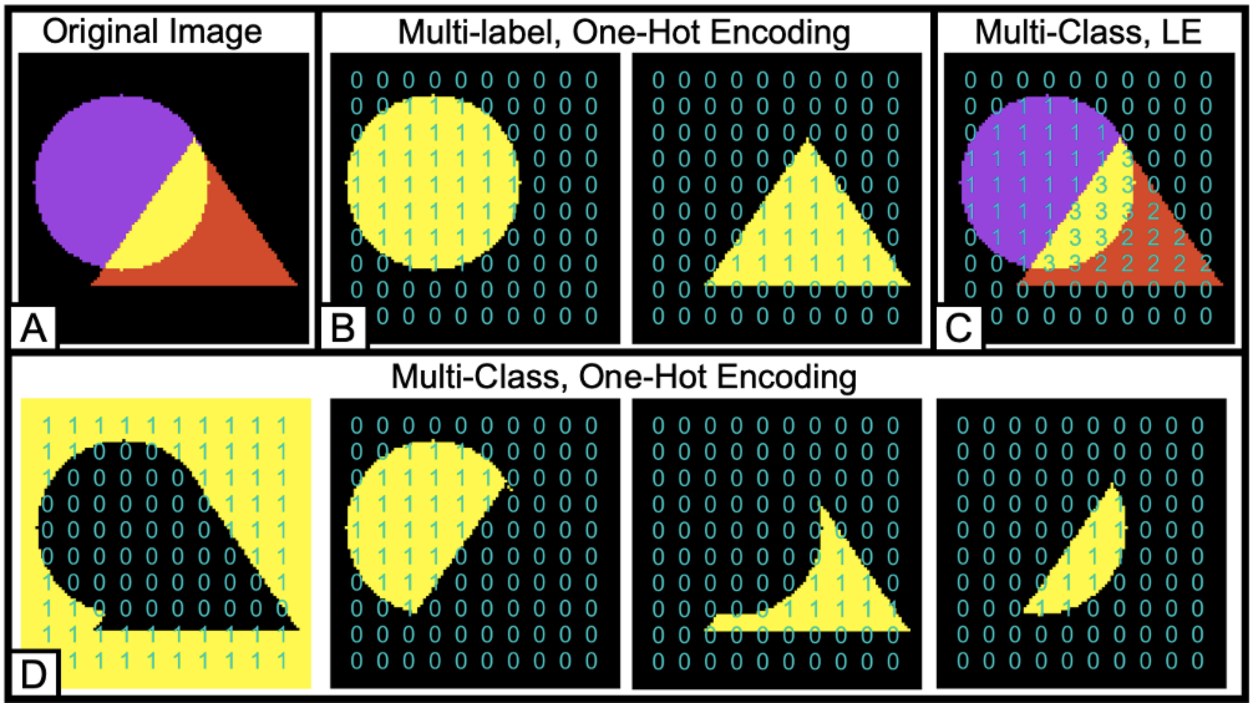


Figure 2.3: A) The original image of a circle and triangle sharing an overlapped region is shown. B) A one-hot encoded multi-label representation of image A. C) A multi-class label encoding (LE) representation of image A. D) A one-hot encoded multi-class representation of image A.

The output of a neural network needs to match the dimensionality of the target ground truth labels. For segmentation, this requires a special output layer to convert the regression from the network into class probabilities for each voxel in the input. Multi-class segmentation requires a softmax function, which is a scaled activation which maps the neural network to a normalized distribution function representing the per-channel estimation of class membership (the sum of the classes for a given voxel predication is equal to one). Despite the output of a softmax activation being normalized, the model output should not be confused with a probabilistic (i.e., Frequentist or Bayesian) output for class membership. This means that probabilistic statistical tests or utilizing a probabilistic determination to inform clinical decisions is not a valid interpretation of a network’s output. Instead, during inference, each voxel has a class assigned to the channel with the highest value, typically by applying a maximum argument (argmax) function, ensuring each voxel is a member of only a single class. However, during model training, the loss is computed from the raw outputs (without the argmax function applied) to compute and backpropagate the gradient of the error with respect to all possible classes.

For a model to achieve multi-label segmentation, the model should conclude with a sigmoid function as the final activation. This ensures that the model outputs normalized, class-independent, per-voxel class membership predictions. Since the sigmoid function is independent for each output channel, a voxel having membership in multiple classes is a valid prediction. Then, during inference, a sigmoid activated prediction is rounded to the nearest binary value, allowing each voxel the potential of being a member of multiple classes. And, similarly to multi-class segmentation training, the loss function should be computed on the raw, or unrounded, predictions.

2.4 Evaluating a Loss Function

In the Appendix A, I discuss many differing loss functions and their applications. With the numerous loss function choices, picking a starting point can be overwhelming. To help choose an initial loss function, I have included a decision tree (Figure 2.4) to narrow down the selection process. But, to get the most out of the chosen loss function, a user should understand how to evaluate and tune the loss function’s performance.

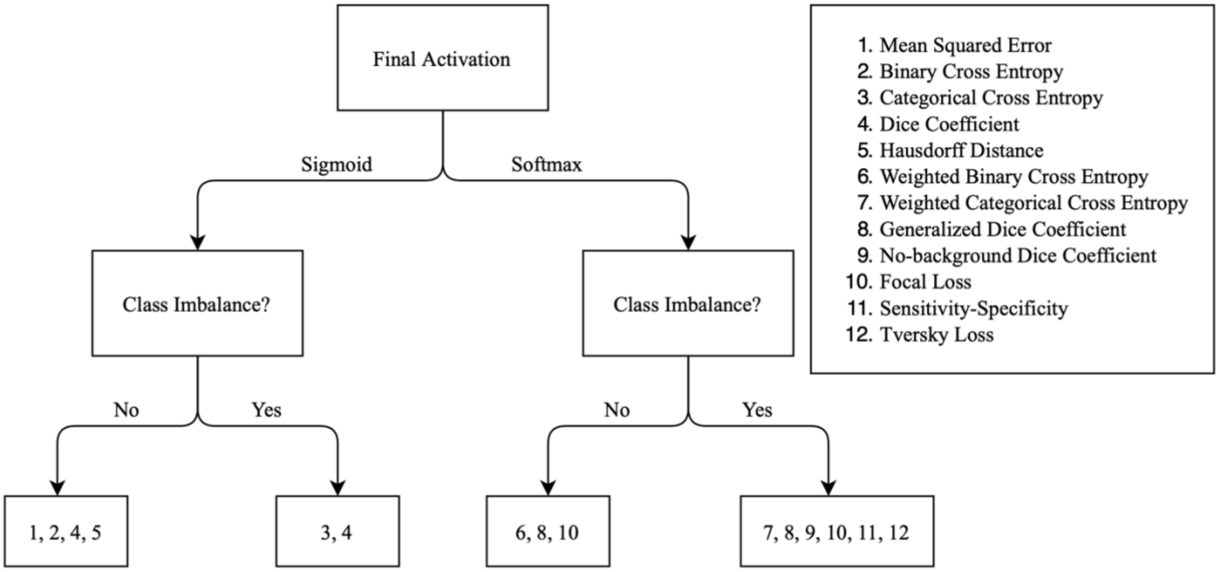


Figure 2.4: A flowchart to aide in determining the proper loss function for a given task.

Typical deep learning strategy dictates a dataset be separated into three unique subsets: training, validation, and testing. The training set, as the name implies, is used to train the model and is the largest of the three subsets. During the training process, predictions made from this data are used for backpropagation weight updates. Following every epoch, the training model makes predictions from a smaller subset of data, the validation set, where predictions are made without updating the model's weights. It should be repeated that deep learning models are lazy and will take whatever shortcuts are available. Commonly, this shortcut is memorization. When a model memorizes, it begins to perform outstandingly on the training data set without learning generalizable features, which means it cannot replicate this performance equally on an unlearned dataset, such as the validation set. By frequently predicting on the validation data set, we can monitor the model's progress in real-time and prevent wasting time when the training is non-ideal. Typically, the relationship of training and validation loss falls into one of four categories, as shown in Figure 2.5.

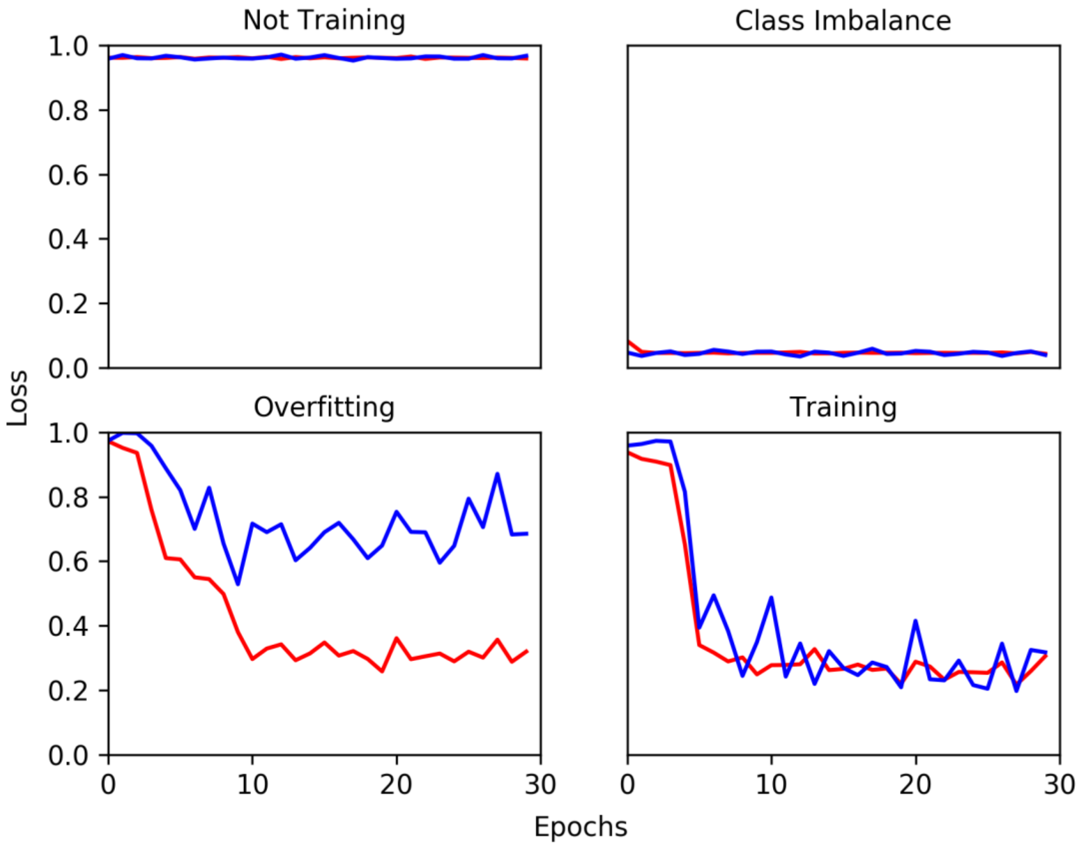


Figure 2.5: A representation of different types of relationships between the training loss (red) and validation loss (blue). A) A model which does not train. B) A highly imbalanced data set with a poorly suited loss function. C) A model which overfits on the training set. D) A model which trains.

A model that consistently performs poorly on both losses across all epochs, as seen in Figure 2.5.A, is indicative of a model that is not training. Unfortunately, there is no clear-cut reason why a model does not train, but troubleshooting should progress through the training process. Beginning with the data, this issue may arise from training data or ground truth labels that are incorrectly formatted or not properly corresponding. Within the model, errant graph connections or incorrect final activation and loss function pairings can prevent the model from properly backpropagating the gradient. Finally, hyperparameters may be poorly selected, causing weights to change too quickly or coarsely to successfully converge to the minima.

A model which immediately produces outstanding and desirable results, like that shown in Figure 2.5.B, is indicative of a highly unbalanced task paired with an unbalanced loss function. At the start of training, a model’s weights are randomly initialized, and are never expected to perform perfectly after only a few iterations of the training cycle. This behavior is typically characterized by a model becoming trapped in an overwhelming local minimum, such as predicting one class for the entire volume. This can be troubleshot through experimentation with alternative loss functions.

An overfitting model, as given in Figure 2.5.C, has a loss function that consistently decreases while the validation loss remains unchanged. To prevent overfitting, common techniques may be to introduce dropout into the model or utilizing optimizer regularization. Additionally, the training data can be augmented to simulate a more diverse dataset.

When everything comes together, and a deep learning model learns properly, we expect both loss functions to decrease relatively steadily and asymptotically to the same value, as shown in Figure 2.5.D. It is important to note that the rate of convergence will vary based on task, model, and optimizer. In this instance, the model was able to learn a generalizable feature from the training data and perform equally well on the validation set. The possibility exists, however, that the chosen loss function is not indicative of desired performance. To check this, the model’s predictions on the validation set should be compared to the ground truth with additional metrics. If these metrics also indicate strong performance, a final prediction on the test set can be created.

For a deep learning model to converge upon a generalizable solution, the method in which it gauges performance, the loss function, must be carefully chosen. Because this loss function quantifies the fitness of the model’s predictions, the loss function dictates the backpropagation process, and in turn how a model learns. While educated guessing may assist in selecting a loss function, finding the ideal function typically requires experimentation with different loss functions or combinations. We described the most popular loss functions within this chapter, but there exist many niche functions which were not discussed. As techniques for medical image segmentation evolve, pioneering individuals will continue to develop novel loss functions capable of greater admissibility and ease of trainability.

2.5 Hyperparameter Tuning

While deep learning model will statistically converge upon a solution as it trains, to achieve the best possible results still requires human input and intuition. The most notable role for humans, aside from model design, is the process of hyperparameter tuning. Hyperparameters are tunable variables in the model training system, most commonly in the loss function, optimizer, and the length of model training. The modification of these parameters alters for how long, how slowly and with what characteristics a model navigates the loss function space.

2.6 Model Evaluation

Traditionally the dataset used for training and evaluating a model is split into three portions: a training set, a validation set and a testing set. As the name implies, the training set is used to saturate the model and train its parameters. The model will see this data set repeatedly, with each full pass across the data set called an ‘epoch’. For each prediction made upon a training set data point, the loss is computed and backpropagated into the model parameters, training the model how to generate more accurate predictions. At the end of each epoch, the model performance can be tracked by comparing quality of prediction on a dataset the model has not been trained upon, which we refer to as the validation data set. This validation data set allows the practitioner to track the model performance on “unseen” data and ensure the model is not simply memorizing the training dataset. Validation data is also useful for trained model selection and the comparison of different model designs, allowing the user to pick the model most capable of predicting robust and accurate results. Although the validation dataset is not used to directly train the deep learning model, through the hyperparameter tuning process, it becomes easy to select model parameters which overfit on the validation dataset. Therefore, it is necessary to have a third dataset held out until all hyperparameters are tuned. This third dataset is called the test dataset and a model is supposed to generate a single, final inference upon this dataset. By only having seen the test set a single time, these data points replicate the actual performance of the model on new and previously unseen data. Because the validation and test sets are typically smaller than the training dataset, it is important that these two subsets are selected carefully to ensure they accurately represent the diversity of the proposed application.

An alternative means of validating robust performance across a dataset is cross-fold validation. During cross-fold validation, a dataset is split into N-constituent parts, say 10-parts where 8/10 are designated for training, 1/10 for validation, and 1/10 for model testing. Then, hyperparameter tuning and testing would be conducted upon that data split. To achieve convergence across the cohort, the hyperparameters would again be held constant and the training and testing data would be reorganized into each of the unique combinations of the data set split. In total, the number of trained models (M) is given by $M = N * (N - 1)$. When conducting the data splitting, the validation set can be handled in two ways. The first is that the validation set can also be shuffled through each of the possible combinations, giving the number of models trained as defined by $M = N * (N - 1) * (N - 2)$. But, considering that some model architectures can require 10s of hours, or more, to adequately train, shuffling through all $N - 2$ times more combinations could require weeks or months more computational time. Additionally, after completing hyperparameter tuning, the role of the validation set is small and poses little benefit in the overall analysis of the model performance. Therefore, it is not strictly necessary to conduct a complete iteration through all possible data set permutations.

CHAPTER 3 Hippocampal Avoidance and Treatment Planning

3.1 Existing Paradigm

Motivated by the high number of untreated brain metastases found in cancer patients upon autopsy, whole brain radiotherapy (WBRT), the complete irradiation of the brain parenchyma, was proposed as treatment for metastatic disease [5]. In the 1954 paper, Chao *et al.* [5] stated that because brain metastases most often occur in multiples, the only logical treatment is to irradiate the entire brain to reduce missing small asymptomatic or sub-resolution lesions. Therefore, Chao *et al.* recommended the use of WBRT in all palliative patients with brain metastases to reduce symptoms and improve survival rates. With the limited technology at the time, the recommended treatment energy was 250 kV, used to deliver a total dose of 3000 rads (rads reported for historical accuracy, 1 rad = 0.01 Gy). This prescription dose was chosen to deliver as close to 2000 rads to the midline without inducing moist skin erythema (dry skin erythema was expected). Treatment fractions were started at 50-100 rad/day and increased by 50 rad/day up to 350 or 400 rad/day, or until headaches began occurring [5] (likely due to acute encephalopathy [6]).

Following the initial proposal of WBRT, a search for the ideal fractionation schedule began, with accelerated fractionation schemes of 1500 rad in two fractions [6] and 1000 rad in a single fraction [6] both proving unsuccessful. In fact, in the single fraction trial, 3 of 54 patients died within 48 hours of treatment due to cerebral edema and hemorrhaging [7]. It was eventually decided that concurrent corticosteroids and treatment fractions 3 Gy or less could reduce the risk of cerebral edema [8]. A Phase III clinical trial investigated an array of treatment schedules and settled upon 30 Gy in 10 fractions to minimize adverse neurocognitive side effects, and increasing the palliative index (survival time in a neurologically improved state) [9].

In the 68 years since its proposal, the contemporary indications to WBRT have remained nearly unchanged. Though mercifully, our treatment planning and delivery has improved. Contemporary treatments no longer determine fractionation limits by induced headaches and dry skin erythema. Instead, WBRT of 30 Gy in 10 fractions is most often used when microscopic or gross disease is present, or targeted chemotherapeutics prove ineffective. Other uses for WBRT include prophylactic brain irradiation for small cell lung cancer and pediatric craniospinal irradiation.

Despite the relative simplicity of WBRT, it has proven clinically effective, even for patients who present with few metastases. For patients who receive stereotactic radiosurgery for one to four brain metastases, the addition of up-front WBRT reduces the brain tumor recurrence rate by 29.6% [10]. Additionally, a trial showed that WBRT after surgery or SRS for one to three metastases reduced the 2-year relapse rate by 32% at the initial site and 10% at new sites [11]. While this trial was successful at reducing the 2-year relapse rate, it failed to meet the primary end point of increased time of functional independence, as measured by a WHO performance status greater than 2 (10 months without WBRT, 9.5 months with WBRT). This paradox of a decreased disease but no improvement in functional status indicated that WBRT caused neurotoxicity and reduced a patient's neurocognitive function. The declines in neurocognitive function induced by WBRT were found to be 31-57% at 3 months and 48-89% at one year [12].

3.2 Why Avoid the Hippocampus?

The hippocampus is a small, seahorse shaped structure located in the medial temporal lobes of the cerebrum, proximal to the temporal horn of the lateral ventricles. Hippocampal involvement in the formation of memories has been known since 1957 when two patients received a bilateral medial temporal-lobe resection and were described to have suffered from “a grave loss of recent memory" [13]. It was noted that the patients suffered no appreciable changes in personality or intelligence, only an acute loss in active memory and partial retrograde amnesia.

Building upon these case reports of lobotomized patients, more sophisticated evidence was uncovered for the role and method in which the hippocampus contributes to the formation of memories. Progenitor cells were discovered to be contained within the subglanular region of the hippocampus dentate gyrus and these cells were shown to contribute to neurogenesis [14]. It is known that the vertebrate brain continually produces neurons and these newly formed neurons contribute to the formation of trace-memories [15]. Therefore, radiation induced damage to the progenitor cells is theorized to inhibit neurogenesis in the dentate gyrus, ultimately impacting the formation of memories and executive function [16]. To validate this theory, clinical trials were undertaken to determine the neurocognitive impacts of hippocampal avoidance during whole brain radiotherapy.

3.3 Proof Through Clinical Trials

3.3.1 RTOG-0933 Phase II

Radiation Therapy Oncology Group (RTOG) Trial 0933 investigated the efficacy of WBRT with hippocampal avoidance (HA-WBRT) [17]. The trial was designed as a single-arm study with historical studies as the control. Adult patients with English proficiency and who presented with metastatic disease more than 5 mm outside of the hippocampus, a nonhematopoietic malignancy (excluding small-cell or germ cell cancer) were eligible for trial enrollment. Patients were excluded if a contraindication for MR imaging existed.

Patient cognitive function and health-related quality of life (QOL) were the primary study end points. To assess cognitive function, the Hopkins Verbal Learning Test-Revised (HVLT-R) [18] was conducted at baseline and 2-, 4- and 6-months post-treatment. The HVLT-R tasks patients with memorizing 12 nouns, then recalling the words immediately and after a 20-minute delay. For the test, patients must also identify the 12 nouns from a list of semantically related or unrelated nouns. These three components of the HVLT-R are designed to assess a patient's cognitive abilities for total recall, delayed recall, and immediate recognition. Patient QOL was evaluated with the Functional Assessment of Cancer Therapy – Brain (FACT-BR) [19] and Barthel Index of Activities in Daily Living (ADLs) [20] questionnaires. A patient's well-being in five categories (emotional, physical, social, functional and brain tumor specific factors) was quantified with the FACT-BR and the Barthel Index of ADLs was used to evaluate the patient's ability to independently complete daily living tasks (e.g., feeding, bathing, dressing). A per-patient relative decline in assessment scores were tracked for each follow-up date, with the baseline used as a control. Of the collected metrics, the primary end point was the HVLT-R delayed recall, for which the historical control found a 30% mean (41% standard deviation) relative decline at 4-months relative to the patient baseline [21].

For HA-WBRT treatment planning, patients were required to receive a 3D T1-weighted axial MR image with axial slice thickness $\leq$ 1.5 mm and a planning CT image with axial slice thickness of $\leq$ 2.5 mm. The MR image would then be co-registered to the CT volume and the aligned secondary MR image would be used for the delineation of the hippocampus. Manual segmentation of the subglanular zone of the hippocampus is defined by hypointense grey matter on the T1-weighted MR images. Per the protocol guidelines, the inferior border of the hippocampal contour is the medial extent of the temporal horn. From there, the hippocampal contour continues to follow the hypointense grey matter superiorly along the edge of the ambient cistern, with the contour terminating when the hypointense grey matter separates from the atrium of the lateral ventricle [22]. The bilateral hippocampal contours were and expanded by 5mm to generate a hippocampal avoidance region. To generate the PTV, the hippocampal avoidance region was subtracted from a contour of the brain parenchyma.

A prescription dose of 30 Gy in 10 fractions to the PTV was used, matching the historical control study [21]. Hippocampal dose constraints of $D_{100} = 9\ Gy$ and $D_{MAX} = 16\ Gy$ were pre-protocol, with $D_{100} \leq 10\ Gy$ and $D_{MAX} \leq 17\ Gy$ as acceptable deviations. To achieve this level of dose contrast in the treatment plan, IMRT treatment planning were required. Centralized rapid review was utilized during enrollment, with sites of three consecutive acceptable enrollments exempt from future pre-treatment review.

RTOG-0933 implemented many of the clinical trial best practices to ensure minimal protocol deviation including: a contouring workshop, creation of a contouring atlas, pre-enrollment credentialing, pre-treatment centralized quality assurance and post-treatment plan review. Despite these measures, accurate hippocampal segmentation during trial enrollment proved difficult, with 26% (26/100) patients having unacceptable clinical contours [23].

This trial enrolled a total of 113 patients, 100 of which were analyzable. For the primary endpoint of HVLT-R at 4-months, 42 patients were analyzable. Among these 42 patients, cognitive decline, as measured by the HVLT-R delayed recall, was found to be 7.0% (95% CI: -4.7% to 18.7%), significantly lower than the historical control of 33.3% cognitive decline at the 4-month follow-up. Total recall performance was also greater, with HA-WBRT resulting in 3.6% (95% CI: -2.9 to 10.1) decline relative to the historical control of 19.0% decline. Scores from the FACT-BR showed significant improvement of the emotional category (p=0.042) and no decline of the other categories relative to baseline. Barthel Index of ADLs follow-up time points showed no improvement or decline when compared to baseline.

Prior to the trial, one of the primary concerns of HA-WBRT was the risk of disease progression in the surrounding hippocampal avoidance region. This trial found that of the 67 patients who developed intracranial disease progression, only 4.5% (3/67) of cases had disease progression in the hippocampal avoidance region. Overall, the trial found a substantial improvement of neurocognitive toxicity sparing through the inclusion of HA-WBRT with only a small increase in risk for disease progression. Therefore, this trial was considered a success and was selected for continuation as a Phase III, multi-institutional trial with control.

3.3.2 NRG-CC001 Phase III

From the positive results in the Phase II trial, the NRG Oncology group formed the CC001 task group to conduct a Phase III HA-WBRT trial [24]. Trial protocols expanded upon the design of RTOG-0933 with the inclusion of a control arm and administration of prophylactic memantine to both arms of the trial.

Glutamate stimulation of the N-methyl-D-aspartate (NMDA) receptor is correlated with degenerative neurological disorders, like Alzheimer’s Disease [25]. Memantine is an NMDA receptor inhibitor which prevents receptor overstimulation and has been shown efficacious in reducing neurocognitive decline in Alzheimer’s Disease [26]. Further studies demonstrated prophylactic Memantine usage concurrent with WBRT to significantly reduce cognitive function failure (53.8% compared to 64.9% control) [27]. As Memantine use during WBRT had become the standard of care, both arms of the NRG-CC001 trials would receive the drug during treatment.

The primary objective of the trial was to determine if HA-WBRT increased the time to neurocognitive failure at specified time points (2, 4, 6 and 12 months). Neurocognitive function and quality of life were again measured as the primary end points. These factors were assessed using three tests: the Hopkins Verbal Learning Test-Revised (HVLT-R) [18] to evaluate total and delayed recall and delayed recognition; Controlled Oral Word Association (COWA) [28]; and the Trail Making Test (TMT) Part A and B [29]. New to this trial, the COWA test gave participants a word category and provides them 60 seconds to verbally state words belonging to that category, thereby assessing a participant's spontaneous word production ability. The Trail Making Test was used to evaluate a patient's visual attention (Part A) and ability to switch between tasks (Part B). Part A of the TMT test provided patient with a sheet of paper with dots numbered 1 to 25 and participants were tasked with connecting the dots, in order, as quickly as possible. Part B modified the dot ordering by including both numbers (1-13) and letters (A-L) and the dots were to be connected in an alternating order (1, A, 2, …, L). All tests were conducted prior to treatment to determine a per-patient baseline score.

Quality of life and severity of symptoms was assessed throughout the trial using the EuroQol 5-dimension, 5-level (EQ-5D-5L) [30] and the MD Anderson Symptom Inventory-Brain Tumor (MDASI-BT) [31] tests. The EQ-5D-5L is a descriptive system designed to measure five dimensions (self-care, usual activities, mobility, pain and depression or anxiety) on a 1-5 scale to quantify the level of problem (none, slight, moderate, severe, and extreme). MDASI-BT is a submodule of the MDASI specific to patients with brain tumors and is a questionnaire designed to determine the severity of 9 brain tumor specific symptoms: astasis, dysarthria, seizures, hemiparesis, difficulty concentrating, problems with vision, changes in appearance or bowl movements and irritability.

Using this battery of tests, the primary end point of the study was to evaluate time to neurocognitive failure, defined as a consistent decline as measured by at least one test. Test scoring was performed by a qualified neurocognitive chair who was blinded to study arm assignment.

Secondary end points include progression free survival, overall survival, toxicity, quality of life and patient-reported symptoms.

Patient enrollment was limited to adult patients with a Karnofsky performance score ≥ 70, without hydrocephalus, leptomeningeal metastases, prior WBRT or the ongoing usage of other NMDA antagonists. Patients with prior surgical resection or stereotactic radiosurgery were eligible for enrollment.

Contouring guidelines and planning requirements remained unchanged from RTOG-0933. Pre-enrollment credentialing and rapid pre-treatment central review was again utilized in this trial. In this trial, if the initial treatment plan was deemed acceptable, all subsequent plans would be reviewed post-treatment only to assess plan quality and establish a channel of ongoing communication.

In total, 518 patients across 112 institutions were randomly assigned to either study arm between July 2013 to March 2018, with nearly equal populations analyzable for WBRT (n=261) and HA-WBRT (n=257). Analysis showed that across the cohort, cognitive failure risk was significantly lower for HA-WBRT (hazard ratio = 0.76; 95% CI: 0.60 – 0.98), confirming the efficacy of HA-WBRT to reduce cognitive decline. Specifically, HA-WBRT showed significantly less decline when measured by TMT Part B (23.3% v. 40.4%; p=0.01) and the 6th-month follow-up data point of HVLT-R total recall (11.5% v. 24.7%; p=0.049) and delayed recall (16.4% v. 33.3%; p=0.02).

Analysis of secondary end points also showed that at the 6th-month follow-up, patients assigned to HA-WBRT experienced reduced symptom interference (p=0.008) and fewer cognitive symptoms (p=0.01). No significant difference was found between treatment arms for percentage of deceased, overall survival (6.3 v. 7.6 months; p=0.31) or intracranial progression free survival

(5.0 v. 5.3 months; p=0.21). As was seen with RTOG-0933, recurrence within the hippocampal avoidance region was unlikely, occurring in 4.3% (11/257) of WBRT and 6.1% (16/261) of HA-WBRT patients.

The results of the NRG-CC001 confirmed the results of the RTOG-0933 trial and supported the hypothesis that conformal avoidance of the progenitor cells contained within the subglanular zone of the hippocampus reduces the neurocognitive failure of patients treated with WBRT. The findings of this Phase III trial have changed standard of care to HA-WBRT combined with prophylactic memantine for patients with brain metastases with an expected survival of ≥ 4 months. This change in the standard of care will necessitate that upwards of 200,000 radiation oncology patients per year receive a high-resolution MR imaging study prior to treatment planning to facilitate the manual segmentation of the hippocampus. For patients contraindicated for MR imaging or institutions unable to obtain MR imaging in a timely manner, without an alternative to manual hippocampal segmentation, the proven cognitive sparing benefits of HA-WBRT will remain inaccessible to patients.

CHAPTER 4 Hippocampal Segmentation with Deep Learning

4.1 Introduction

Whole-brain radiotherapy (WBRT) is the most widely used treatment for patients with multiple brain metastases. Past studies have shown that patients whose tumors regress due to radiation treatment experience increased quality of life due to improved neurocognitive function (NCF) [32]. However, studies also found that WBRT is associated with an early decline in NCF, particularly deficits in learning, memory, and spatial processing [16]. The hippocampus is a paired structure located in proximity to the temporal horn of the lateral ventricle and is critical to the process of memory formation. Radiation-induced injury to the hippocampus is known to alter learning and memory function [33–36].

Hippocampal avoidance during WBRT treatment (HA-WBRT) has been investigated as a means to prevent hippocampal injury and subsequent NCF toxicity with decreased quality of life [17,37]. Studies demonstrate that sparing the hippocampus from radiation without altering the coverage of the rest of the brain decreases early NCF decline without compromising disease control. Dosimetric studies utilizing various techniques to spare the hippocampus including both intensity modulated radiotherapy (IMRT) and helical TomoTherapy treatment first demonstrated the feasibility of hippocampal avoidance during WBRT [22,38–41]. RTOG 0933 was a multi-institutional phase II trial of HA-WBRT for brain metastases [17]. In RTOG 0933, only 7% (8 / 113) of patients experienced decline in memory as compared to historical controls with 30% experiencing NCF decline when irradiated without hippocampal avoidance. There was no decline in quality of life scores in study patients versus significant decline in historical controls [42]. A phase III study (NRG CC001) has been completed and reported at national conferences [43,44], demonstrating that conformal avoidance of the hippocampi during WBRT preserves NCF while achieving similar intracranial control and survival.

HA-WBRT requires accurate delineation of the hippocampus. In each of the HA-WBRT studies to date, identification of the hippocampus is performed on high resolution T1-weighted MRI images [45] fused to radiation treatment planning computed tomographic (CT) images of the head [23]. With an MRI, sufficient soft tissue contrast exists to delineate the hippocampus. The MRI is then fused to a CT, reducing geometric distortions inherent in MRIs, but fusion can possibly introduce other sources of error [46,47]. For example, differences in the axial spacing of the slices introduces errors from interpolation. In the RTOG 0933 multi-institutional trial, prior to participation in the study, treating physicians were individually credentialed, where an example patient MRI and CT volumes were fused, contoured and planned, with the results then reviewed by the RTOG centralized committee [23]. A Hausdorff distance [48] of > 7 mm between physician's contours and the reference contour or errors in MRI-CT fusion were considered unacceptable deviations. For the trial, failures were significant, with 6.8% (8 / 113) of physicians failing credentialing and 15.85% (13 / 82) enrollees failing pre-treatment centralized review due to errors in either MRI-CT fusion or hippocampal segmentation.

Deep learning provides a method to train computational models with the representations needed for object detection or classification [3,49]. Deep convolutional networks, one class of models inspired by visual neuroscience [50], have achieved breakthrough success in the detection, segmentation, and recognition of objects in images [51–53]. Convolutional operations are foundational to contemporary deep learning segmentation models. Since the first implementation with LeNet, the field has exploded with creative solutions to segmentation tasks [54], such as encoder-decoders [55–57], residual connections [58] and inception blocks [59]. Model complexity and input tensor dimensions have been limited by the need to fit the entirety of the data within the Graphics Processing Unit (GPU) memory. The hippocampus is a 3D structure whose segmentation is best performed using volumetric (3D) image data. Fortunately, as GPU on-board memory and speed have increased, 3D convolutional networks have emerged, such as 3D U-Net [60] and 3D ResNet [61].

The field of neuroimaging has explored 3D deep convolutional network models for brain structure segmentation, including hippocampal segmentation [61,62], but none have attempted segmentation from CT images directly without using MRI as input. Zhao *et al.* [63] demonstrated the feasibility of CT based segmentation by using a 2D deep learning model to generate synthetic MR images, after which a deformable atlas registration was used to segment structures from the synthetic MRI, including the hippocampus. Since Zhao *et al.*, deep learning models have been shown to outperform deformable atlas-based segmentation techniques for hippocampal segmentation [64], and 3D neural networks have been shown to outperform 2D networks for both direct segmentation tasks [60] and synthetic image generation [65]. Further spurred on by the development of deep learning specific computational cards, particularly Nvidia's tensor cores, models have become deeper [66], more computational intensive [67,68], and expanded into three dimensions, either spatially [60] or temporally [69,70]. Additional tools have been co-opted from other deep learning fields for segmentation purposes, most recently attention gates. Attention gating was first utilized for natural language processing [71,72] to direct the attention of deep learning models towards relevant words in a sequence. Attention gating has since been utilized for super resolution [73], image classification [74], 2D image segmentation [75] and volumetric medical image segmentation [76,77].

Three-dimensional deep learning models have the potential to automate hippocampal segmentation and remove the need for additional MRI scans. As demonstrated by the credentialing experience in RTOG 0933, removing the need for the MRIs to identify the hippocampus will reduce the need for a second imaging study for treatment planning, reduce the potential uncertainties associated with MRI-CT image registration, and reduce the cost and complexity of treatment. The vast diversity of developments in deep learning models and methodologies has provided a wealth of tools to improve patient outcomes in radiation oncology.

HA-WBRT has the potential to benefit approximately 200,000 patients per year in the United States alone [78,79]. In this study, we demonstrate that deep learning models, utilizing 3D convolutional neural networks, can delineate the hippocampus using only high-resolution non-contrast CT images, with accuracy comparable to human physicians on a national randomized trial.

4.2 Methods

4.2.1 Image Data

Under a Beaumont Research Institute Institutional Review Board approved retrospective study (2018-009), we collected high resolution CT and MRI images acquired for Leksell Gamma Knife (Elekta AB, Stockholm, SE) radiosurgery treatment planning. During treatment planning, each patient had a stereotactic frame placed by a neurosurgeon. Following placement, sequential, high-resolution imaging studies were conducted using 16-slice Siemens Sensation 16 CT scanner (Siemens Medical Solutions, Malvern, PA) and a gadolinium contrast enhanced T1-weighted sequence on a 3T Siemens Sonata MRI scanner (Siemens Medical Solutions, Malvern, PA). In total, 402 Gamma Knife patients were visually inspected and those with significant artifacts or anatomy-altering tumors (e.g., meningiomas) were excluded. Of those inspected, 390 patients were selected for this study. The selected cohort was either healthy brain (trigeminal neuralgia or vestibular schwannoma; 191 patients) or treated for metastatic disease (4 to 26 brain metastases; 199 patients), with treatments between July 16, 2007, and July 19, 2018. Images suitable for this study were then collected in a MIM workstation (MIM Software Inc., Beachwood, OH). Using MIM, the MRI volumes were rigidly registered and resized to the coordinate space and voxel dimensions of the CT volume (MR resampled from $1.0 \times 1.0 \times 1.0$ to $0.5 \times 0.5 \times 1.0$ mm). By using the fiducial markers on the stereotactic frame during image alignment, sub-millimeter accuracy in the rigid registration was achievable [80].

4.2.2 Contouring the Hippocampus

Contours of the hippocampus were created following the methods and guidelines of Chera *et al.* [45] and Gondi *et al.* [22] using thin-sliced (< 1.5 mm slice spacing) T1-weighted MRI images. The hippocampus contouring tutorial atlas from the RTOG 0933 study was used as a reference for contouring consistency [81]. Following contour generation, a minimal smooth operation was applied, and the final contours were reviewed for anatomic accuracy.

4.2.3 Image Processing

In addition to the hippocampal contours, a body contour was generated using a threshold region grow tool. This body contour was used to mask out the Gamma Knife frame, preventing a deep-learning model from learning on the frame’s integrated spatial fiducial markers. Removing the frame also produced trainable image volumes which more closely resembled conventional WBRT simulations, which do not utilize a stereotactic frame.

These patient CT images and structure sets containing the left and right hippocampus contours were anonymized and transferred to a research server for further processing. From the body contour, a center of mass was calculated for each patient and a global offset was applied to center the cropped volume at the level of the hippocampus. Image volumes for each patient were cropped to $200 \times 200 \times 35$ voxels centered around a standard offset determined across the entire dataset. Cropping was used as opposed to down-sampling to reduce the theoretical loss in segmentation accuracy caused during the down, and subsequent up-sampling process. A short investigation of these impacts of resampling are presented in Appendix B. All contours were converted from DICOM format into a binary mask for the left and right hippocampus. Then, the processing operations were repeated in an equivalent manner to the segmentation masks.

To simplify and expedite training, the CT image was processed three ways for use in model training: soft-tissue window-level, bone window-level and an inverse-square distance map computed from the calculated center of mass (bottom row, Figure 4.1).

The soft-tissue window and level was computed from the HU values within the cropped image volumes across the dataset. A Gaussian distribution curve was fit to the HU histogram, from which we included $\pm 4\sigma$ to maximize dynamic range. The resulting soft-tissue window and level had values of 80 and 56 HU. For the bone window-level, the standard settings in the MIM software suite were used, with a window of 2800 HU and a level of 600 HU.

One method to reduce memory requirements is patch-based image segmentation, which segments small image patches, with results determined by majority voting. The small field of view (FOV) of these training patches can be difficult to learn from due to the low context and contrast of brain CT images. To maintain the relative spatial information of these small patches, providing a relative coordinate system can improve performance [82–84]. In practice, during training a deep learning model should learn relationships between spatial regions and image features, although it cannot be assumed for all model designs and domains [85,86]. But, akin to the challenges presented by small FOV patches, the initial convolutional operations in a neural network are limited to the kernel size (typically 3x3x3). Therefore, without explicitly providing spatial information, it can be difficult for these initial convolutional operations to derive any meaningful detail from their limited FOV on a low-contrast cranial CT image. Even in instances where spatial relations are developed, the initial convolutional operations may be used ineffectively if few identifiable features can be learned. So, to ensure efficient convolutional operation utilization, we provide a channel of the input tensor which is an inverse-square distance map with the distance measured from the center of mass (Figure 4.1.F).

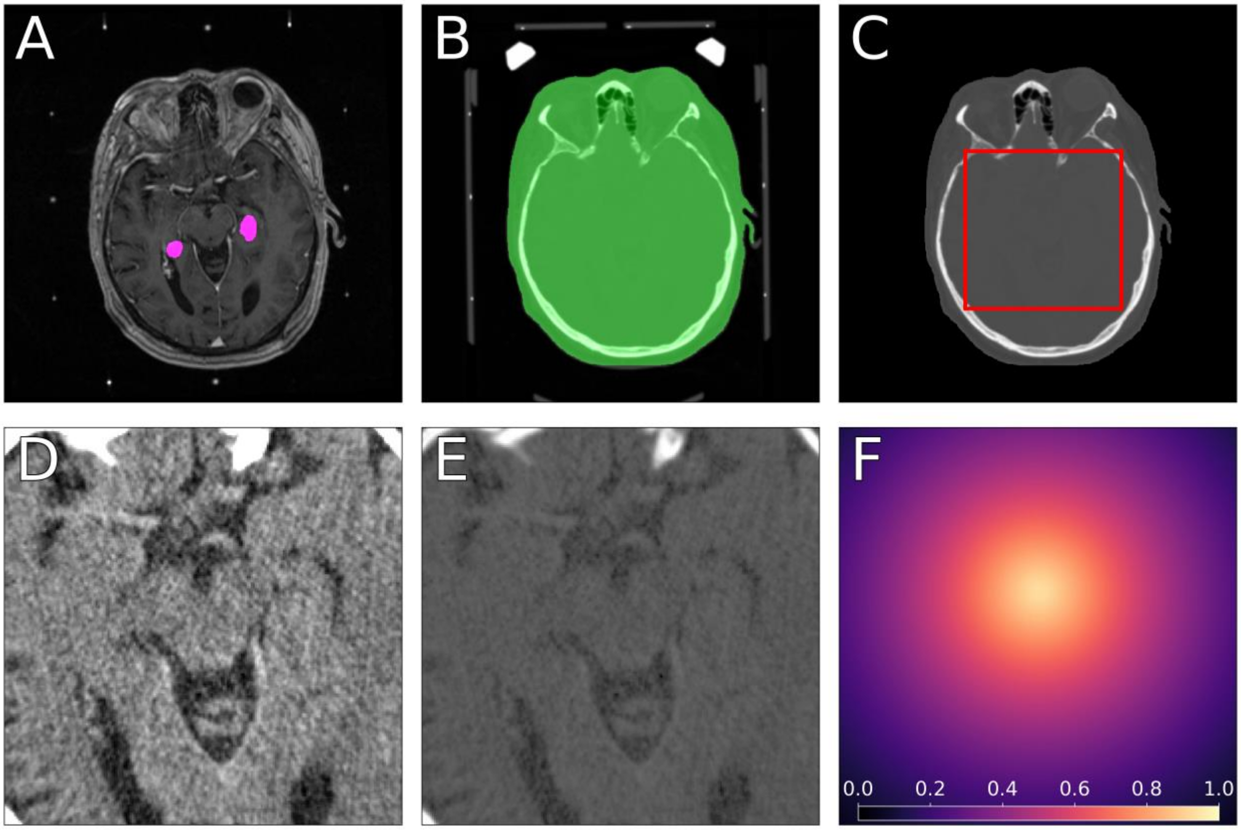


Figure 4.1: Visualization of data processing steps (first row) and inputs (second row). A) MRI with ground truth (magenta) B) CT Image with body contour (green) C) CT Image with frame masked out, red box indicating cropped volume D) Cropped volume window, leveled to soft-tissue E) Cropped volume window, leveled to bone F) Inverse squared distance map from centroid.

4.2.4 Model Design and Training

Radiation oncology treatment simulation utilizes helically acquired CT scans which are reconstructed into a 3D image volume, from which anatomical structures can be segmented. A 2D deep learning model is not ideal for deep learning segmentation in this domain because it predicts each slice independent of surrounding slices, making the model inherently prone to predictions with disjointed surfaces or incongruencies which almost always perform worse than 3D models.

In this investigation, we only considered 3D deep learning models with a large enough field of view to segment the entire hippocampus at native resolution. This prevented the need to implement a work-around for a smaller FOV utilized in other models, such as two stage models [87,88], predicted volume up-sampling or down-sampling [68,89–91], small FOV sliding window inference [92], and conditional random fields [89–91], all of which may limit model performance for our segmentation task.

To develop the best model for our proposed segmentation task, we compared three existing models that have been utilized for brain segmentation tasks. Then, motivated by the specific needs of our task, we propose a fourth model of novel design. The three existing models tested include the 3D U-Net [60] (Figure 4.2, top), the Dilated 3D U-Net [93] (D-3D U-Net; Figure 4.2, bottom), and the High-Res3DNet [61] (3D ResNet; Figure 4.3, top). Our novel model is the Attention Gated 3D ResNet (AG-3D ResNet; Figure 4.3, bottom). For each model, hyperparameters were individually tuned and then their performance was compared using a nested cross-fold validation across our entire dataset.

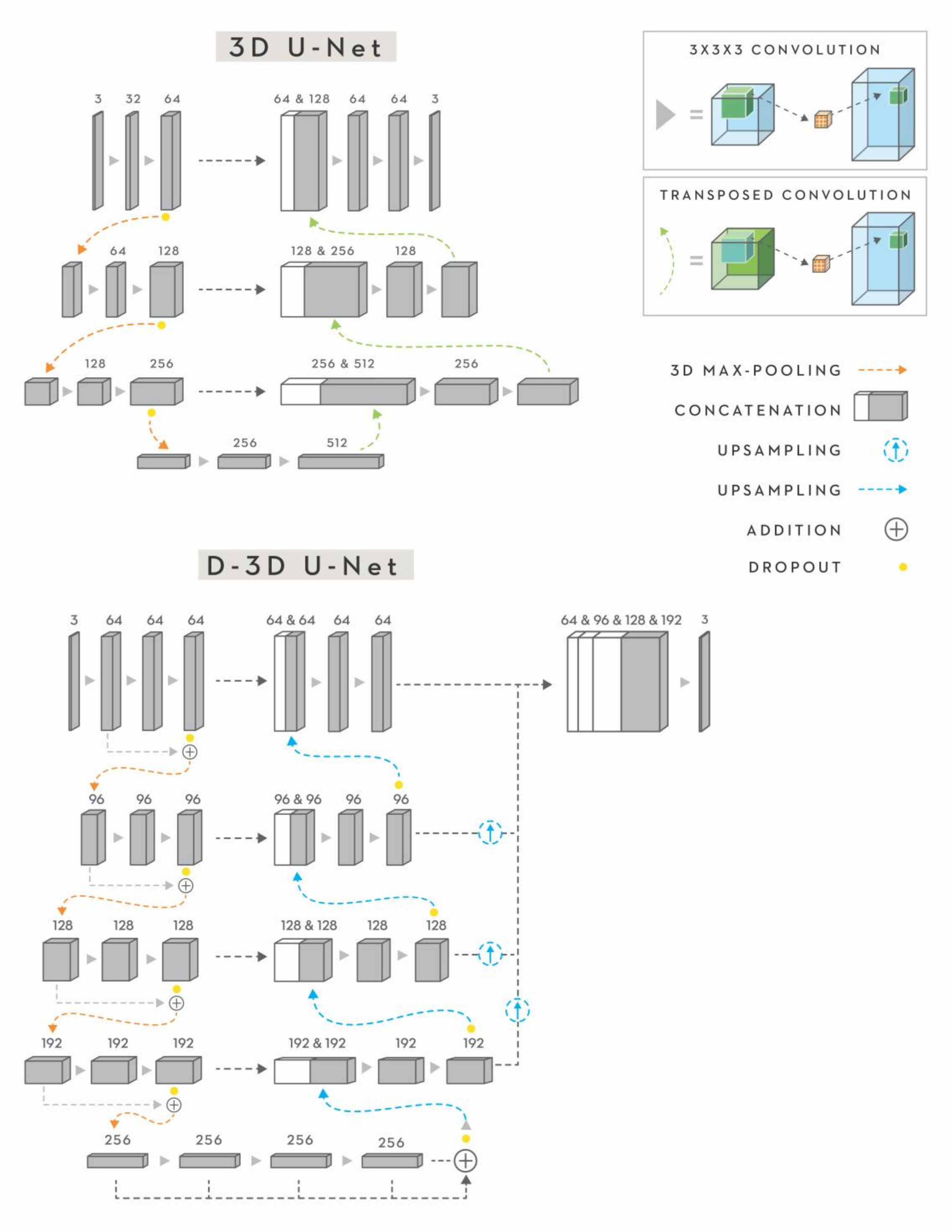


Figure 4.2: A depiction of the 3D U-Net (top) and D-3D U-Net (bottom) models, where the 3D U-Net includes transposed convolutions during decoding and the D-3D U-Net uses 3D Up-sampling operations. The D-3D U-Net cascaded output matches image dimensions before addition with up-sampling as well. The number of convolution filters per operation is noted with a number above the convolutional operation or block.

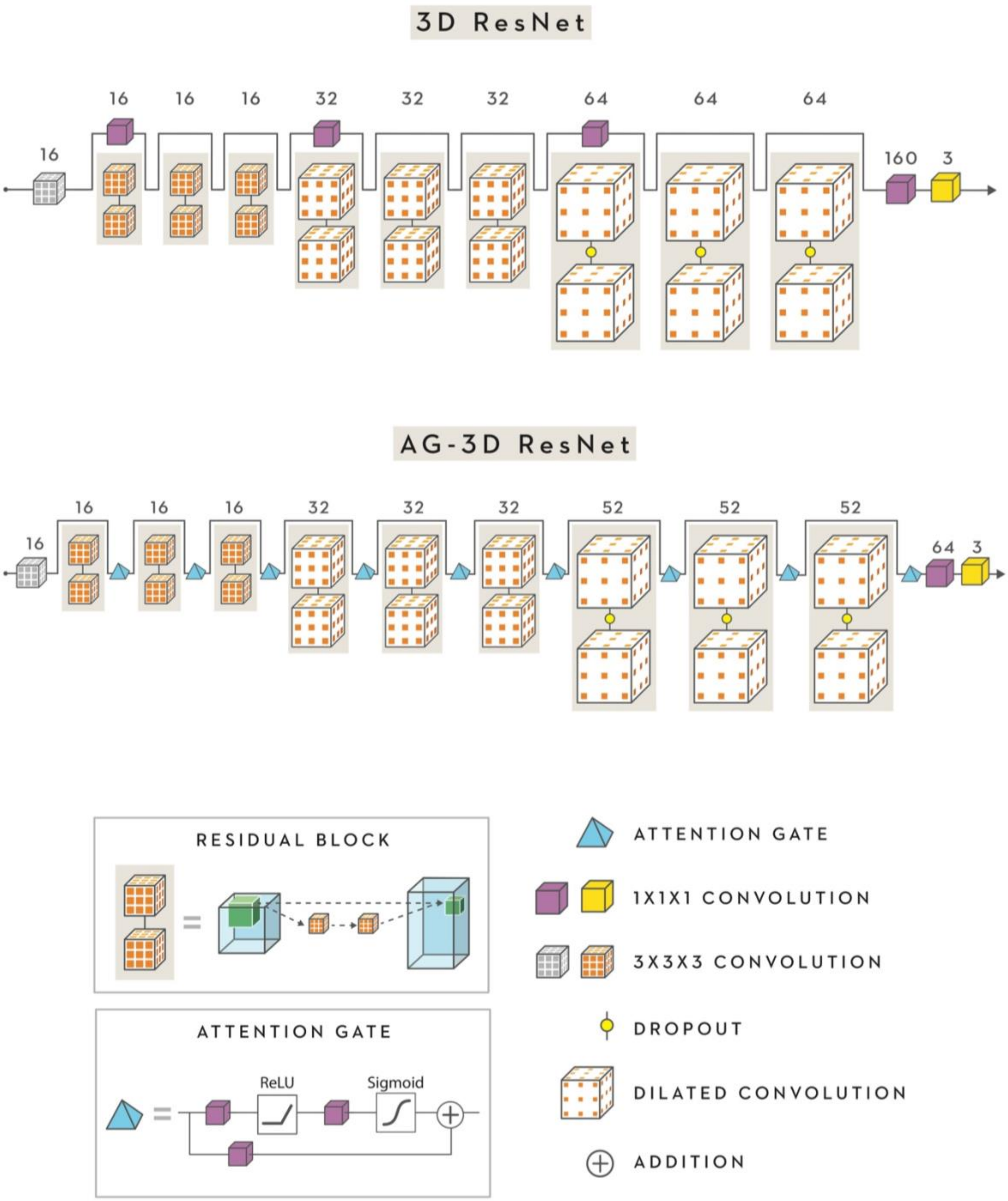


Figure 4.3: A depiction of the 3D ResNet (top) and AG-3D ResNet (bottom). For both models, residual connections include a convolutional operation to match the number of filters prior to residual or attention gating addition. The number of convolution filters per operation is noted with a number above the convolutional operation or block.

For the 3D U-Net we chose to implement the standard model, for which the model design and properties have been discussed elsewhere [60]. Folle *et al.* [93] proposed the D-3D U-Net for hippocampal segmentation on MR images. This model is a derivative of the U-Net design which replaces the lowest U-Net layer with a summation of four dilated convolutions, adds short residual connections in encoding blocks, replaces the transposed convolutions with 2D up-sampling operations, and generates the final prediction with a cascaded summation of up-sampled outputs from each decoding layer. In our implementation of the D-3D U-Net, we modified the number of convolutional filters to 64, 96, 128, 192, and 256 for each layer, respectively. By decreasing the filter sizes, we allowed for increased input dimensions, at no apparent decrease in model performance. The High-Res3DNet model is an implementation of 3D ResNet which allows for a larger field of view on the training input [61]. We optimized the base model design by altering the location of dropout layers and adding an additional layer to the decoding structure.

For most deep learning problems, most of the training data is irrelevant to accurately solving the task. Derived from natural language processing, attention gates were developed to focus the model to reinforce high yield regions of the training data such as nouns or verbs. From natural language processing, the application of attention gating has improved performance in segmentation tasks [72–77,94]. During manual hippocampal segmentation, the lateral ventricles and white matter dictate most of the contour's borders, suggesting that only a small portion of the image is critical to generating accurate segmentations. Motivated by this realization, we propose a novel model architecture called the Attention-Gated 3D ResNet (AG-3D ResNet), which introduces additive attention gates [76,94] in the residual blocks. During experimentation in model design, we found additive attention gating to significantly outperform multiplicative gates. While the inclusion of additive attention gating in residual blocks impedes the gradient back-propagation [95], the difference does not prevent saturation of a model as small as the AG-3D ResNet. In exchange for decreased training speed, additive attention gating reinforces regions of particular interest, which aids in segmenting small, low contrast structures. To maintain a comparable memory footprint to the 3D ResNet, the last set of the model's residual blocks were decreased from 64 to 52 filters, and the second to last block was decreased from 160 to 64 filters.

Input to each model was a three-channel tensor comprised of the soft-tissue and bone window-levels, and an inverse-square distance map (Figure 4.1.D-F). Both the 3D ResNet and AG-3D ResNet models were trained with the three-channel tensor at dimensions $200 \times 200 \times 35$ voxels. Due to the down- and up-sampling of the 3D U-Net design, the input tensor was limited to dimensions with a factor of two, so the tensor was cropped to $192 \times 192 \times 32$. The D-3D U-Net's increased memory footprint necessitated further limiting the input tensor to $192 \times 192 \times 16$, which was randomly generated from within the 3D U-Net's training dataset during training. Both the $200 \times 200 \times 35$ and $192 \times 192 \times 32$ included most of the hippocampus voxels (> 99%). For each model design, the output tensors had equivalent dimensions to the input tensors. For the D-3D U-Net, final volumes were inferred from sets of three predictions, with majority voting used to resolve the overlapping region of the volume. In all model designs, the final layer culminated with a softmax activation, generating three channels corresponding to left hippocampus, right hippocampus, and background (neither). The number of model parameters and input tensor dimensions are given in Table 4.1.

Table 4.1: Models with number of parameters and input tensor dimensions

MODEL	**AG-3D RESNET**	**3D RESNET**	**DILATED 3D U-NET**	**3D U-NET**
# OF PARAMETERS	644,535	830,339	14.14 million	19.08 million
INPUT TENSOR SIZE	(200, 200, 35) x 3 channels	(200, 200, 35) x 3 channels	(192, 192, 16) x 3 channels	(192, 192, 32) x 3 channels

Hyperparameters were determined by training 10 models across the same train, test, and validation split, optimizing until the model consistently performed well. The hyperparameters for each model are provided in Table 4.2. A nested cross-fold validation followed, which, due to the high number of trainings required, was parallelized across four Nvidia Titan RTX GPUs (Nvidia, Santa Clara, CA) with 24 GB memory and two Nvidia Quadro RTX 8000 GPUs (Nvidia, Santa Clara, CA) with 48 GB memory. All models were trained on a 312 / 39 / 39 split for train, validation, and test, respectively.

Table 4.2: Hyperparameter settings used for each model during nested cross-fold validation.

MODEL	**AG-3D RESNET**	**3D RESNET**	**DILATED 3D U-NET**	**3D U-NET**
BATCH SIZE	2	2	2	2
MAX EPOCHS	25	15	30	25
LEARNING RATE (LR)	2.5E-4	2E-4	7.5E-4	7.5E-4
OPTIMIZER	ADAM	ADAM	ADAM	ADAM
LR DECAY	0.0	0.0	2E-8	2E-8
DROPOUT	0.175	0.15	0.25	0.25
LR REDUCTION	0.25 After 2 epochs	0.50 After 2 epochs	0.5 After 3 epochs	0.25 After 2 epochs
EARLY STOPPING	After 5 epochs	After 3 epochs	After 5 epochs	After 4 epochs

For data augmentation during training, we generated images with transformations randomly chosen between ± 10 mm x, y-axis shifts, ± 2 mm z-axis shifts, $\pm 10°$ rotation (roll) and a 50% likelihood of inclusion of between ±5% gaussian noise and 50% likelihood of flipping along the y-axis. During training of the D-3D U-Net, random 16 slice sub-volumes of the U-Net data set were generated during training. We found data augmentation to not significantly change overall training results ($p > 0.25$), which is likely attributable to our large and homogenously sourced dataset with single institution origin.

4.2.5 Loss Function and RTOG Evaluation Metrics

The accuracy and clinical applicability of segmentation in radiation oncology is dependent on both the similarity and maximum spatial separation between the predicted and ground truth contours. The most common metric used to determine segmentation similarity is the Dice similarity coefficient [96]. When using a Dice loss function [97] for tasks with large class imbalances, a model may tend towards segmenting only the largest volume class. To account for class imbalance, the generalized Dice loss [98] scales the per-class Dice loss by the relative class occurrence in the ground truth. For hippocampal segmentation, the background is orders of magnitude larger than the hippocampi and the generalized dice loss imbalance would be substantial. To simplify the training, we took the limit of the generalized Dice loss for a large background by simply excluding the background channel altogether. This exclusion meant we instead calculated the Dice loss from only the left and right hippocampus. To facilitate an equal comparison, we utilized this loss function when training all model designs. Coincidentally, excluding the background for the dice loss provides an accurate metric for model checkpointing, early stopping, and determining learning rate updates while training. While the Dice similarity score is robust and easily interpretable for determining the relative spatial agreement, absolute spatial disagreement is important in radiation oncology treatment planning. For this reason, RTOG 0933 utilized an acceptance criterion based upon the Hausdorff distance metric, given in Equation 4.1, which calculates the absolute spatial disagreement between two contours. The RTOG 0933 trial protocol determined a HD $\leq$ 7 mm as an acceptable deviation.

$$\text{Hausdorff}(X, Y) = \max\{\sup_{y\in Y} \inf_{x\in X} d(y,x), \sup_{x\in X} \inf_{y\in Y} d(y,x)\} \tag{4.1}$$

In addition to Dice and Hausdorff, we also compared Jaccard score, average surface distance, relative average volume difference, precision and recall between the predicted segmentation and corresponding manual hippocampus structure.

4.2.6 Volume Inference and Nested Cross-Fold Validation

A well performing deep learning model should be able to robustly segment the structure of interest on any given patient. But, with a single train / test data split, the test set is unlikely to fully represent the dataset domain. Furthermore, a model may be unable to consistently learn features when trained repeatedly. This may either be due to a propensity to overfit, susceptibility to local minima or an incapability of consistently learning to identify features. To evaluate the true performance of our models, we performed a 10-fold nested cross validation which totaled 90 trained instances of each model, each with 39 predicted volumes per test split, for a total of 3510 predicted volumes per model type. Through the cross-validation process, we reduce any variances introduced by initial patient shuffling and splitting for a test set. Furthermore, the overall mean across all folds is computed from the entire 390-patient dataset, giving a more representative picture of model performance across a large cohort. In total, we trained 90 instances each of the 3D U-Net, D-3D U-Net, 3D ResNet and AG-3D ResNet models, requiring approximately 46 GPU days to train the four models when parallelized across four Titan RTX GPUs and two Quadro RTX 8000 GPUs. On both card types, test set inference occurred in less than one second per volume.

4.3 Results

4.3.1 Comparing Deep Learning Results to Physicians on RTOG 0933

Across our entire cross-fold validation (3510 predicted volumes), the AG-3D ResNet generated predictions for left and right hippocampus which achieved a mean and standard deviation Hausdorff distance of 4.78 ± 2.53 mm and 4.63 ± 2.20 mm. This translated into an RTOG passing rate of 88.3 ± 31.4% and 88.9 ± 32.1% for left and right hippocampus. During the pretreatment centralized review of the RTOG 0933 trail, 82 patients were enrolled, with the treating physician contours having a mean Hausdorff distance of 5.47 mm. On the first attempt, 13.41% (11 / 82) failed for hippocampal segmentation and 2.44% (2 / 82) failed for MRI-CT fusion for a combined failure rate of 15.85% (13/82). Because our workflow forgoes the need for MRI-CT fusion, we compared our results to the RTOG population which passed on both contour and registration evaluation. Considering that for the RTOG 0933 study, 6.8% (8 / 113) of physicians failed credentialing prior to patient enrollment, it is reasonable to assume the passing rate observed in the study is at least representative of the average clinical radiation oncologist.

To mirror the workflow observed in the clinical trial where only bilateral hippocampi were compared, the model predictions for each patient were only considered passing if both the left and right hippocampi met the RTOG 0933 criteria independently. The total passing rate of the AG-3D ResNet was 80.2%, (2815 / 3510), which when compared to the RTOG 0933 study using a two-sided t-test, the null hypothesis could not be rejected (p = 0.3345). For the other model designs, the 3D ResNet likewise could not reject the null hypothesis (p = 0.1677), whereas both the 3D U-Net or D-3D U-Net performed significantly differently (p < 1E-5) from the RTOG-0933 trial.

Using a two-sided Wilcoxon signed-rank test, the 90 trained model instances for each design were compared based on the RTOG passing criteria. We found both the ResNet style models to outperform either U-Net style model (p < 1E-5), and the AG-3D ResNet to significantly outperform the 3D ResNet (p = 0.045). Both the D-3D U-Net and 3D U-Net experienced complete failure rates (no hippocampus was predicted), as indicated in Table 4.3, while neither the AG-3D ResNet or 3D ResNet had any such failures. These failures are evident in the boxplot of the passing rates for the 3D U-Net (Figure 4.4) where an entire quartile failed for right hippocampus.

To determine the impact of the inverse square distance map, we re-trained and tested the AG-3D ResNet with the distance-map channel replaced with zeros. The trained instances without the distance map were found to have performed significantly worse by both 95% Hausdorff distance and Dice score (p<0.05). Additionally, without the inverse square map, the AG-3D ResNet model experienced the only instances of complete failure to segment either hippocampus. For any such failing hippocampi predictions, the 100%, 95% Hausdorff distances and average surface distance are undefined and are indicated in Table 4.3 as ‘INF’. To aide in visualizing the model predictions, a segmentation from each model is displayed in Figure 4.5. In this example, both U-Net models failed to predict one hippocampus each.

Table 4.3: Metrics reported as median and interquartile range comparing AG-3D ResNet, 3D ResNet, Dilated 3D U-Net and 3D U-Net. Bolded text indicates the best performing model for the given statistic, determined by mean squared error from the ideal value.

COMPARISON METRIC	**AG-3D RESNET**		**3D RESNET**		**D-3D U-NET**		**3D U-NET**	
	Left	**Right**	**Left**	**Right**	**Left**	**Right**	**Left**	**Right**
DICE SCORE (%)	**73.8 (68.8–78.5)**	**73.7 (68.5 – 77.8)**	73.0 (67.7 – 78.0)	72.7 (67.7-78.0)	66.6 (57.2 – 72.1)	65.4 (53.6 – 70.9)	70.7 (60.4 – 76.9)	68.0 (00.0 – 74.6)
JACCARD SCORE (%)	**58.5 (52.5 – 64.6)**	**58.3 (52.1 – 63.6)**	57.4 (51.1 – 63.9)	57.1 (50.8-62.7)	49.9 (40.1-56.4)	48.5 (36.6 – 54.9)	54.7 (43.3 – 62.5)	51.5 (00.0 – 59.5)
HAUSDORFF (MM)	**4.062 (3.162 – 5.523)**	**4.153 (3.240 – 5.500)**	4.123 (3.202 – 5.612)	4.272 (3.240 – 5.679)	5.612 (4.039 – 8.559)	5.679 (4.123 – 8.768)	4.822 (3.500 – 8.031)	5.500 (3.742 – INF)
95% HAUSDORFF (MM)	**1.803 (1.414 – 2.449)**	**1.871 (1.414 – 2.449)**	1.871 (1.414 – 2.500)	2.000 (1.500 – 2.500)	2.236 (1.803 – 3.669)	2.291 (1.871 – 4.243)	2.121 (1.500 – 3.905)	2.500 (1.803 – INF)
AVERAGE SURFACE DISTANCE (MM)	**0.548 (0.421 – 0.738)**	0.551 (0.431 – 0.713)	0.557 (0.433 – 0.761)	**0.551 (0.438 – 0.720)**	0.645 (0.477 – 0.993)	0.645 (0.499 – 0.982)	0.644 (0.462 – 1.059)	0.713 (0.503 – INF)
RELATIVE ABSOLUTE VOLUME DIFFERENCE (%)	**10.2 (-8.2 – 31.6)**	**9.1 (-9.9 – 31.5)**	9.0 (-10.7 – 31.9)	6.9 (-12.9 – 30.1)	4.8 (-26.0 – 34.4)	1.0 (-31.9 – 32.9)	4.2 (-27.6 – 31.0)	-8.9 (-100 – 22.7)
PRECISION	**0.723 (0.626 – 0.806)**	0.722 (0.632 – 0.797)	0.720 (0.621 – 0.807)	**0.722 (0.689 – 0.853)**	0.637 (0.502 – 0.742)	0.631 (0.491 – 0.725)	0.671 (0.517 – 0.777)	0.623 (0.000 – 0.756)
RECALL	**0.798 (0.707 – 0.859)**	**0.786 (0.701 – 0.857)**	0.783 (0.689 – 0.853)	0.773 (0.672 – 0.849)	0.716 (0.560 – 0.815)	0.695 (0.502 – 0.804)	0.762 (0.585 – 0.856)	0.694 (0.000 – 0.819)
FAILURE RATE (%)	0.00	0.00	0.00	0.00	13.3	14.4	16.7	30.0
RTOG PASSING (%)	**88.3**	**88.9**	87.1	87.7	64.5	66.4	70.8	59.7
BILATERAL RTOG PASSING (%)	**80.2**		78.5		44.6		39.5	

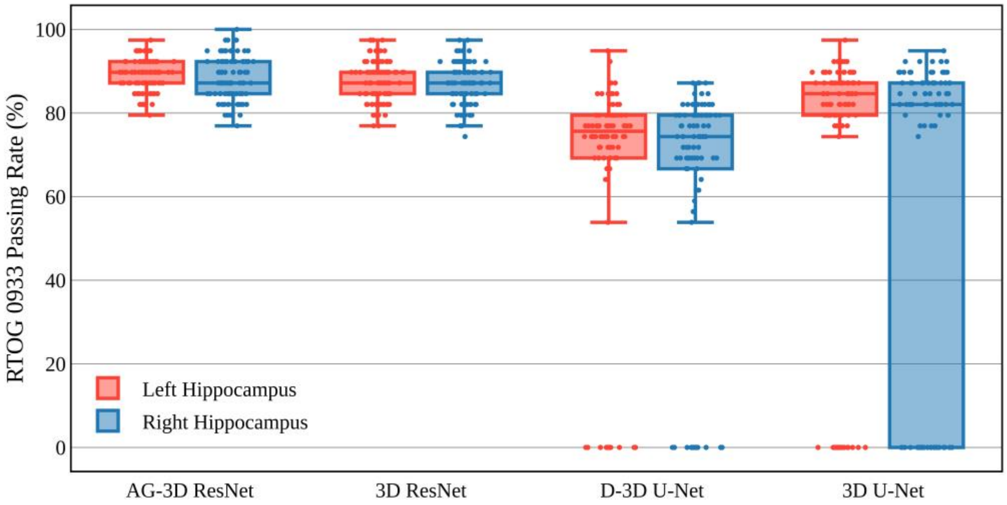


Figure 4.4: Passing rates for left and right hippocampus for all four models. Points at 0% passing represent a trained instance where the model fails to predict one of the hippocampus volumes. Best viewed in color.

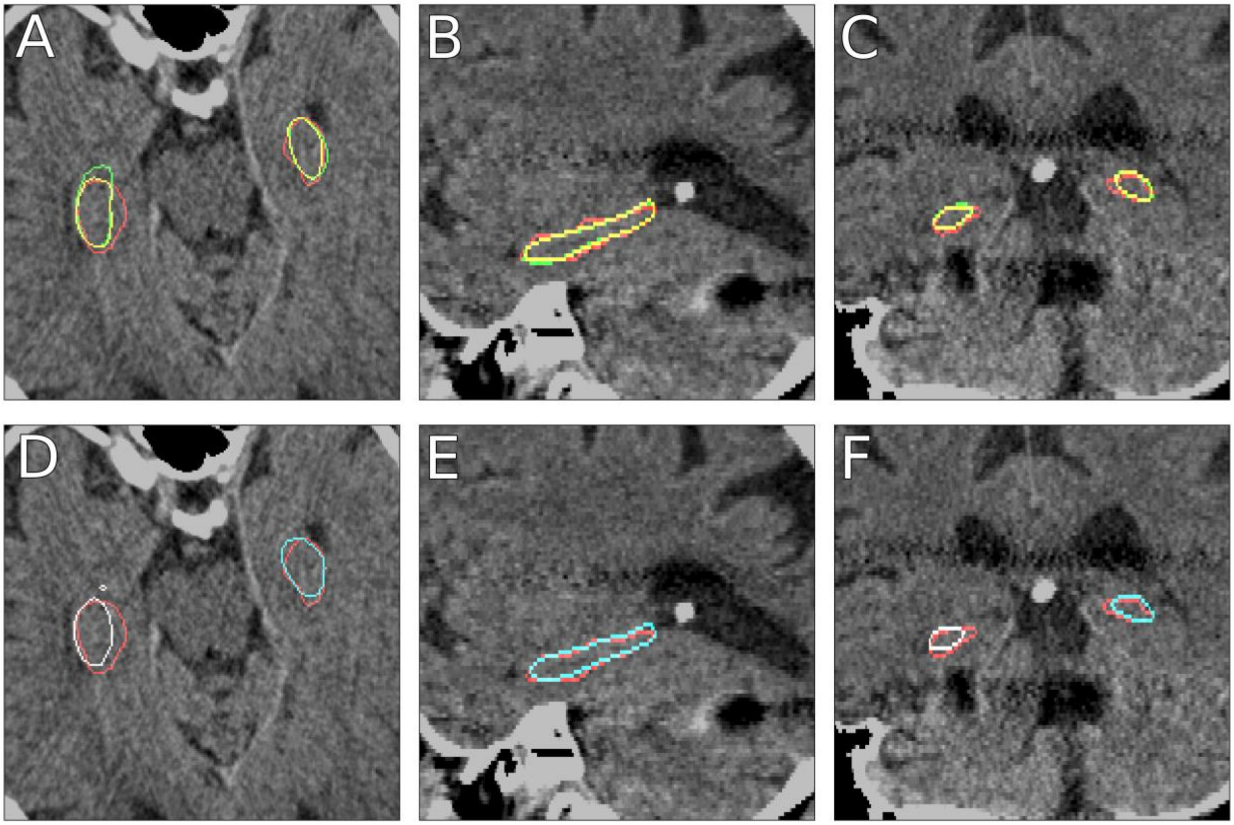


Figure 4.5: Visual comparison of hippocampus segmentation, shown on axial (A, D), sagittal (B, E), coronal (C, F). Contours are Ground Truth (red), AG-3D ResNet (yellow), 3D ResNet (green), D-3D U-Net (cyan) and 3D U-Net (white). Note that sub-figures D and F show an example where both the D-3D U-Net and 3D U-Net failed to predict one of the hippocampus volumes; right and left, respectively. Best viewed in color.

4.4 Discussion

4.4.1 Comparing Models

Most of the performance difference between the 3D ResNet based and 3D U-Net based models is attributable to the 3D U-Net's propensity to overfit. While we cannot scientifically conclude that the absolute optimal hyperparameters were chosen, and equivalent hyperparameter tuning grid search was used for all models, indicating that the U-Net derived models are more challenging to tune for this task. Although the cascaded output of the D-3D U-Net does partially alleviate the overserved overfitting, the detriment to the overall RTOG criteria pass rate is still significant. Furthermore, the U-Net style models have upwards of 30x more parameters than the 3D ResNets, requiring a larger dataset to fully back-propagate the gradient throughout the model. (Discussed more in depth in Appendix B.2). With a larger number of parameters, when combined with the commonly used ADAM optimizer, the U-Net derived models are sensitive to becoming trapped in local minima during training [99], likely explaining why many of the cross-validation instances only predicted one hippocampi. A common solution to prevent local minima is to instead use the stochastic gradient descent (SGD) optimizer [100] which prevents momentum from trapping the model into local minima. Although, without momentum, the SGD optimizer traditionally requires more epochs to converge, which would have been unfeasible in this study considering the number of model instances that were trained during the cross-fold validation.

As an alternative to the encoder-decoder style models, dilated convolutions with residual connections provide a large field of view with an efficient means of gradient backpropagation. Despite the limited number of parameters, residual networks behave as an ensemble of smaller, individual networks [95], providing an efficient way to encode complex structures. Though the unsampled volumes of the AG-3D ResNet make it less efficient in the usage of GPU memory, the price and amount of on-card dedicated graphics memory has continued to become more affordable.

4.4.2 Hausdorff Robustness for Contour Evaluation

In the reporting for the RTOG 0933 Phase II enrollment results, to investigate the high segmentation failure rates, the trial's administrators investigated the 7 mm HD cutoff and determined it was clinically appropriate [23]. We sought to extend this investigation on the high sensitivity of the 100% HD metric and found that for volumes which failed (695 / 3510), many were failing with discrepancies less than the image voxel dimensions. Of the failures, 23.7% (165 / 695) deviated by less than the distance of one axial voxel (0.5 mm) and 30.1% (214 / 695) deviated less than one voxel diagonally (0.7 mm), shown in Table 4.4.

Table 4.4: RTOG HD Metric Robustness from volumes generated from the AG-3D ResNet

HD MARGIN (MM)	LEFT RTOG PASSING (%)	RIGHT RTOG PASSING (%)	BILATERAL RTOG PASSING (%)	POPULATION DIFFERENCE
0.0	88.3	88.9	80.2	---
0.5	91.1	91.6	84.9	+ 165
0.7	91.9	92.5	86.3	+ 214
1.0	93.3	93.0	88.0	+ 273

The difference of a single voxel in the agreement of two contours is unlikely to manifest in significantly different treatment plans. This highlights the high sensitivity of the 100% Hausdorff distance metric as a stand-alone hard threshold for a clinical trial, and the need for an alternative or combined metric threshold. Furthermore, the RTOG protocol dictates treatment plan optimization should be performed to the 5 mm expansion of the combined hippocampi. To evaluate the agreement of the functional avoidance regions, we re-computed the statistics for the AG-3D ResNet predictions with a 5mm expansion, which are given in Table 4.5. While the Hausdorff distance and average surface distance remain nearly unchanged, the volumetrically sensitive metrics (Dice, Recall, Precision, RAVD) increase substantially.

Table 4.5: Metrics calculated between ground gruth + 5 mm and AG-3D ResNet + 5 mm expansions, reported as median and interquartile ranges. Expansion improves volumetrically sensitive metrics (Dice, Jaccard, RAVD, Precision, Recall), while not improving spatially dependent metrics (HD, 95% HD, ASD).

COMPARISON METRIC	AG-3D RESNET + 5 MM EXPANSION	
	Left	Right
DICE SCORE (%)	87.8 (84.9 – 90.1)	87.5 (85.0 – 89.8)
JACCARD SCORE (%)	78.2 (73.7 – 82.0)	77.8 (73.9 – 81.5)
HAUSDORFF (MM)	4.062 (3.162 – 5.500)	4.153 (3.202 – 5.477)
95% HAUSDORFF (MM)	2.091 (1.732 – 2.915)	2.236 (1.732 – 3.000)
AVERAGE SURFACE DISTANCE (MM)	0.624 (0.483 – 0.817)	0.637 (0.500 – 0.817)
RELATIVE ABSOLUTE VOLUME DIFFERENCE (%)	0.8 (-7.2 – 11.0)	1.2 (-8.6 – 11.7)
PRECISION	0.885 (0.828 – 0.928)	0.883 (0.827 – 0.924)
RECALL	0.896 (0.844 – 0.935)	0.895 (0.837 – 0.935)
RTOG PASSING (%)	89.4	88.4
BILATERAL RTOG PASSING (%)	80.7	

4.5 Conclusion

Within this chapter, an investigation was conducted into the feasibility of using deep learning neural networks for the segmentation of the hippocampus from CT alone. Through the comparison of multiple model architectures, it was determined that the AG-3D ResNet model design yielded the highest and most consistent performance. We found that the segmentations were potentially comparable in protocol compliance to treating physicians on the RTOG-0933 Phase II trial. While this demonstrates the feasibility, the contours used for training and validating were all generated by a single institutional observer. Therefore, validation of the methodology using multi-institutional data is required. In this chapter the applicability and robustness of the 100th-percentile Hausdorff distance metric for the evaluation of clinical comparable contours was brought into question. Due to the sensitivity and hard threshold of the metric, the 100th-perecentile Hausdorff distance may not strongly correlate to the ability to create clinically appropriate treatment plans. We intend to conduct a secondary analysis where a treatment planning study, as opposed to contour comparison, is used as the primary end point of the trial to reduce reliance on the HD metric. From this, we can potentially propose an alternative contour comparison threshold which more strongly correlates to clinically equivalent treatment plans.

The work presented in this chapter was published in a peer-reviewed journal article in Medical Physics in 2020 [101]. Publication of the CT, MR and hippocampal contours used in this investigation is on-going. We expect that the dataset will complete curation and be available via The Cancer Imaging Archive (TCIA) sometime in the second half of 2022.

CHAPTER 5 Methodology Validation Using the RTOG-0933 Dataset

5.1 Introduction

The lack of robust central quality assurance during prospective study of new radiotherapy paradigms leads to poor patient outcomes and may also reduce the statistical power of a study. For example, a post-hoc analysis of a TROG head and neck trial demonstrated a 20% reduction in overall survival for non-protocol compliant plans [102]. Furthermore, a secondary analysis of RTOG 0617 demonstrated that variability in heart contours reduced the power to detect survival decrements from heart dose [103]. While credentialing and centralized pre-treatment quality assurance (QA) minimize protocol deviations [104,105], it is not ubiquitous among clinical trials. A review of 42 clinical phase III trials found that only 45% of trials required credentialing and 52% included pre-treatment review [106].

Radiation induced damage to the neural stem cells have been shown to cause cognitive decline, namely in executive function and delayed recall [37]. Conformal hippocampal avoidance of the subgranular stems cells in the hippocampus during whole brain radiotherapy (HA-WBRT) was shown to reduce the decline in neurocognitive function in a phase III trial [24]. In the phase II feasibility trial (RTOG 0933), the subgranular zone proved difficult to contour, with 6.8% (8/113) of the RTOG 0933 trial participants failing credentialing on the first attempt, 62.5% (5/8) failed on the second attempt, and during enrollment, 26% (26/100) of clinical contours had unacceptable deviations utilizing a criterion of Hausdorff distance (HD) > 7mm from central reviewer contour [17,23]. Contour and treatment plan heterogeneity may have thus reduced the observed benefit of HA-WBRT for some patients.

Previous work has investigated the feasibility of deep learning, and specifically deep convolutional neural networks (dCNNs), for contour quality assurance on MRI [107] and CT [108–110] datasets. Men *et al.* [108] performed lung contour QA by training a network on the 2017 AAPM Grand Challenge dataset and a subset of the RTOG 1308 contours, and evaluated their performance on the remaining RTOG 1308 data. Nijhius *et al.* [110] used a single-institution dataset to contour salivary glands on a subset of the EORTC 1219 dataset. These studies used bootstrapped cutoffs for contour acceptability based on the standard deviation of two agreement metrics: the Dice coefficient and the Haussdorff distance (HD) comparing the treating physician (TP) contours to a manually validated subset with subjectively assessed high-quality contours.

Prior work has demonstrated that a CT-only dCNN hippocampal segmentation model can accurately delineate the subgranular zone hippocampal contours [111]. This study sought to assess such a model's ability to perform contour quality assurance. Specifically, we hypothesized that a single-institution CT-only hippocampus model would achieve a higher compliance with protocol criterion of HD<7mm on the multi-institutional RTOG 0933 dataset compared with the TP contours. We also hypothesized that such a model would provide utility as a first-pass QA tool. Uniquely, we benchmark the model's quality assurance performance for detecting non-protocol compliant treating physician contours against a simulated expert principal investigator- here referred to as institutional observer (IO) – as opposed to subsets of the trial data as performed in prior work.

5.2 Methods

5.2.1 Training Dataset

Images were collected under institutional review board approval for 390 patients treated between 2007 and 2018 at the Beaumont Gamma Knife Center. Of the 390 patients, 192 were treated for metastatic disease of unspecified origin (1-26 lesions) and 198 treated for benign tumors (acoustic neuroma or trigeminal neuralgia). Patients with anatomy altering tumors (i.e., large meningiomas) were excluded from the dataset, but no restrictions on age, sex or prior medical history were made when selecting patients. The CT images were acquired at 120 kVp and variable mAs using either a Siemens Sensation (10, 16 and 64 slice; n=72, 305 and 1) (Malvern, PA) or Siemens Definition AS+ (128 slice; n=12) at a slice thickness of 1 mm and reconstructed to a 21-30 cm axial field of view. T1-weighted, gadolinium-enhanced MR images were acquired using a Siemens Symphony TIM (n=242), Siemens Sonanta (n=143) or GE Signa HDxt (n=5) (Chicago, IL) at a slice thickness of 1mm utilizing a fast spoiled gradient echo sequence, reconstructed to a 25-30 cm field of view. Gamma knife images were chosen for the training dataset because the CT and MR images were high resolution, acquired in back-to-back imaging studies, and could be accurately rigidly registered using the stereotactic frame.

In MIM (Beachwood, OH; version 7.1.3), the T1 MR was rigidly registered to the planning CT. From the aligned secondary MR image, and using the RTOG 0933 contouring atlas as reference, three independent observers contoured the hippocampi (n=390; n=247; n=107) and the contours were saved as RTSTRUCT to the CT frame of reference. Because most patients (n=283) had only two or one observer contours, it was not feasible to compute a consensus contour (e.g., STAPLE [112]) as substantial uncertainty would result in volumetrically smaller contours. The training dataset is expected to be made available on The Cancer Imaging Archive.

5.2.2 Internal Test Set

For use as an internal test set, all cases treated at Beaumont Health which contained bilateral hippocampal contours were collected. In total, 76 clinical cases were identified from at least 6 treating physicians, with treatments between 2013-2021. The CT images were acquired at 120 kVp and variable mAs on a Phillips Brilliance Big Bore (16 slice; n=73) (Cambridge, MA) or a Siemens Sensation Open (24 slice; n=3) at a slice thickness between 1 and 3 mm and reconstructed to a 35-66 cm axial field of view. T1-weighted, gadolinium-enhanced MR images were acquired using the vendor specific fast spoiled gradient echo sequence on either a Siemens (n=48), Philips (n=17), or GE (n=11) MR scanner at slice thicknesses of 1-6 mm and reconstructed to a 16-35 cm field of view. Every hippocampal contour was reviewed prior to usage to ensure contour completeness, but no alterations were made to the original contours.

5.2.3 External Test Set - RTOG 0933 Dataset

The Phase II RTOG 0933 multi-institutional data set was used as a hold-out external test set to validate our model and perform the simulated QA run. Enrollment for RTOG 0933 was limited to patients of age 18 years or older with a Karnofsky performance status greater than 70 and who presented with brain metastases without hydrocephalus, leptomeningeal metastases, or tumors within 5 mm of the hippocampus [17]. For treatment planning, the study protocol mandated T1 weighted MR image with axial slice thickness of at most 1.5 mm, and at most 2.5 mm for the planning CT [17]. The study enrolled 113 patients, 96 of which were provided for this study with complete data including the treating physician contours, T1-MR and CT images.

Prior to enrollment, each physician underwent credentialling which consisted of contouring a trial case, registering the MR to CT, and generating a treatment plan. Following successful credentialling, rapid pre-treatment centralized QA was used for the first three consecutive patients per institution, with subsequent post-treatment review of remaining cases. The RTOG 0933 data set provided for this investigation included only those generated by the treating physician (TP). No alterations were made to the treating physician contours or avoidance volumes in the RTOG 0933 clinical data. As a surrogate for central review contours, an institutional observer (IO; author EP), blinded from existing contours, generated bilateral hippocampus contours on each RTOG 0933 patient following the contouring atlas guidelines. For our simulated first-pass QA, these IO contours would be treated as the centralized reviewer ground truth to evaluate the sensitivity of the deep learning QA tool.

5.2.4 Data Preparation

The cohorts of Gamma Knife (n=390), internal clinical (n=76) and RTOG (n=96) imaging sets were anonymized and collected on a research server. On the research server the images were converted into NumPy arrays using our open-source image processing pipeline [113]. During image processing, any completely or partially missing axial CT image slices within the RTOG 0933 image data set were linearly interpolated. If the interpolated axial image slice resided within a pre-existing contour, the contour was algorithmically interpolated [114]. Each data set then underwent a similar processing pipeline of resampling, cropping, window leveling and normalization. For both test sets the images were resampled to a uniform voxel size ($0.977 \times 0.977 \times 1.25$ mm) using a nearest neighbor function for segmentations and a 3rd order spline function for images. Additional resampling ratios were used for the training set as a form of data augmentation resulting in a range of voxel dimensions between $0.5 \times 0.5 \times 1.0 - 1.5 \times 1.5 \times 2.0$ mm (all voxel sizes given in Table 5.1). Following resampling, each constructed CT volume was window and leveled to both soft tissue (window =375, level = 40) and bone (window = 2800, level = 600) and then normalized. To reduce the model's memory footprint and increase the training convergence, volumes were cropped around a centroid that was calculated on the Gamma Knife data set to maximize the inclusion of hippocampal contours. Maximal hippocampal inclusion was achieved with a cropped volume of 150x175x53 voxels and centroid defined in the axial plane by the center of mass of the skull, offset by 3.53mm posterior, 1.15 patient right and 92.7 mm inferior of the superior aspect of the skull (>100 voxels with $650 < HU < 2000$). Every ground truth segmentation was cropped in an equivalent manner to the CT image volume. The resultant processed volumes were

$146.5 \times 170.6 \times 66.25$ mm and contained 99.998% of the RTOG clinical trial hippocampal volumes and 100% of the single-observer contours. The details of our cohort are summarized in Table 5.1.

Table 5.1: Data set preparation parameters. For RTOG 0933 Test set, contours include treating physician (TP) and institutional observer (IO).

	TRAINING SET	INTERNAL TEST SET	RTOG 0933 TEST SET
PATIENTS	390	76	96
HIPPOCAMPAL CONTOURS PER PATIENT	Up to 3 (different observers)	1 (clinical)	2 (TP, IO)
RESAMPLED IMAGE RESOLUTION(S)	0.5x0.5x1.0 mm 1.0x1.0x1.0 mm 1.0x1.0x1.25 mm 1.0x1.0x2.0 mm 1.5x1.5x2.0 mm	1.0x1.0x1.25 mm	1.0x1.0x1.25 mm
NUMBER OF INSTITUTIONS	1	1	63
CROP SIZE	150x175x53	150x175x53	150x175x53
CROP CENTROID OFFSET (A-P, L-R, S-I)	(-3.35, -1.15, -92.7) mm	(-3.35, -1.15, -92.7) mm	(-3.35, -1.15, -92.7) mm
WINDOW LEVEL (HU)	W: 375, L: 40 W: 2800, L: 600	W: 375, L: 40 W: 2800, L: 600	W: 375, L: 40 W: 2800, L: 600

5.2.5 Model Design and Training

Continuing from a prior work, an attention gated 3D-ResNet (AG-3D ResNet) was used in this study [111]. The AG-3D ResNet model is derived from a 3D residual network design [115] with the addition of attention gates [76] in each residual block. In the interest of simplifying the data processing pipeline, the inverse distance map input used in prior work was excluded from the model inputs. The AG-3D ResNet model was implemented in TensorFlow (version 2.3.0) [116] and consisted of 642,215 trainable parameters. Training utilized a Dice loss function [117], computed excluding the background channel, and was conducted on two Nvidia Quadro RTX 8000 GPUs (Santa Clara, CA) with data parallelism. During training, the processed CT image and segmentation ground truth volume pairs were generated with pitch, yaw and roll rotation randomly chosen between -5 and 5 degrees for each axis. Data generation used a random ordered, parallelized generator where each institutional observer's contour, at each resampled resolution, was generated once per epoch.

Model tuning and training were split into distinct phases. During the first phase, only Beaumont Gamma Knife data was used for hyperparameter tuning, with the data set split into training (n=350 patients) and validation (n=40 patients). A manual hyperparameter grid search was performed to tune learning rate, drop-out rate and learning rate drop to maximize performance on the validation set. The highest performing model parameters are given in Table 5.2. In the second phase, the hyperparameters were held constant and the model instance was re-initialized with random variables and re-trained 5 times on the Gamma Knife data with the same data split. Each trained model instance was evaluated on the internal test set by median Dice coefficient, and the highest performing trained instance was used for final inference on the RTOG 0933 external test data set. Boolean segmentation masks were generated form the inferred predictions on a voxel-by-voxel basis by converting the highest predicted channel (between left, right and background) to one and all other channels to zero. This method was used instead of rounding to ensure each voxel corresponded to exactly one class. Predictions were cleaned-up with a binary fill holes operation and then only the largest contiguous structure was preserved. The final predicted segmentations generated from the RTOG test set were resampled to the original voxel dimensions for statistical evaluation against the unprocessed treating physician and our stand-in for central review contours (institutional observer).

Table 5.2: Hyper-parameters used with the AG-3D ResNet during inference upon the RTOG-0933 dataset.

HYPERPARAMETER	QUANTITY
BATCH SIZE	4
EPOCH	Up to 100
TRAINING STEPS PER EPOCH	876
LEARNING RATE	0.005
OPTIMIZER	ADAM
DROPOUT	0.175

LR REDUCTION	½ After 3 epochs of plateau on validation set loss
EARLY STOPPING	After 8 epochs of plateau of validation set loss

5.2.6 Evaluation

The three sets of contours were evaluated using the MedPy library [118] to compute Dice correlation coefficient [96], 95%, 99% and 100% Hausdorff distance (HD) [119], average surface distance (ASD), relative absolute volume difference (RAVD), precision and recall. During enrollment in the RTOG 0933 clinical trial, contours with a HD < 7mm was defined as within protocol; therefore, we computed how many of the hippocampal volumes would pass this metric. The deep learning predictions generated from the RTOG test set were resampled to their original resolution and processed by filing holes and removing discontinuous voxels. The processed predictions were then compared to the non-resampled contours generated by both the single-observer and RTOG clinical trial. The PTV avoidance volumes for each contour were generated from five-millimeter expansions and the evaluation metrics were repeated.

5.2.7 Treatment Planning

From the test dataset, 32 patients were randomly selected for treatment planning and further analysis. For each of the 32 cases, IMRT treatment plans were generated from each of the institutional observer (IO), treating physician (TP), and deep learning (DL) hippocampal contours. To ensure the plans were clinically applicable, all additionally required contours (optic nerves, chiasm, lens, brain stem and brain) were generated automatically using LimbusAI (Regina, SK, Canada). The generated contours were then manually reviewed and edited by a physician to ensure completeness and accuracy. Treatment plans were generated using RayStation (version 6; RaySearch Laboratories AB, Stockholm) for a commissioned 6 MV beam on an Elekta Versa HD linear accelerator with Agility multi-leaf collimator (Elekta AB, Stockholm). To reduce human bias during planning, an automated script was created to generate and optimize the three plans for each patient. Optimizer dose constraints, given in Table 5.3, were chosen to generate plans compliant with the RTOG-0933 protocol. Planning was conducted in 7 sets of 50 iterations, with an intermediate dose plan generated every 25 iterations and a final dose plan generated every 50 iterations. Dose grid voxel dimensions were set to the RayStation default dose grid size ($3 \times 3 \times 3$ mm) and the grid extent was automatically created to entirely enclose the external contour. The treatment plan isocenter was set to the geometric center of the PTV and four VMAT beam segments (two co-planar arcs, two vertex beams) were created (Table 5.4), with each beam segment limited to 300 second delivery time. After completing all rounds of optimizations, the treatment plan was normalized to the prescription dose ($D_{95\%} = 3000$ cGy).

Table 5.3: Optimization constraints used for the RayStation planning script.

STRUCTURE	CONSTRAINT	WEIGHT
PTV_OPT	$D_{98\%} \geq 2500$ cGy	5
PTV_OPT	$D_{95\%} \geq 3025$ cGy	30
PTV_OPT	$D_{2\%} \leq 3600$ cGy	100
HIPPOCAMPI	$D_{99\%} \leq 900$ cGy	15
HIPPOCAMPI	$D_{MAX} \leq 1600$ cGy	25
EXTERNAL	$D_{MAX} \leq 3750$ cGy	25
OPTIC CHIASM	$D_{MAX} \leq 3000$ cGy	5
OPTIC NERVE (L/R)	$D_{MAX} \leq 3000$ cGy	5
GLOBE (L/R)	$D_{MAX} \leq 3000$ cGy	5
LENS (L/R)	$D_{MAX} \leq 700$ cGy	5

Table 5.4: Beam names and settings used for treatment planning.

BEAM NAME	COUCH ANGLE	GANTRY ANGLE	COLLIMATOR ANGLE
CW	0˚	180˚ to 179˚	45˚
CCW	0˚	179˚ to 180˚	315˚
VERTEX CW	270˚	179˚ to 0˚	5˚
VERTEX CCW	270˚	0˚ to 179˚	355˚

5.2.8 Contour Comparison Statistics

To determine the predictive power of the deep learning (DL) model, a receiver operating characteristic (ROC) curve was created from the DL to RTOG treating physician (DL:TP) Hausdorff distance for the prediction of institutional observer to treating physician (IO:TP) failing contours. The area under the ROC curve was then computed. Additionally, the Wilcoxon signed-rank test was used to compare Dice TP:IO to Dice DL:IO. To investigate the false negative cases, a two-sided Mann Whitney U test was conducted between the true positive and false negative samples comparing the Dice correlation coefficients of the IO:TP contours on both left and right hippocampus. Spearman correlation coefficients were computed for HD to Dice on TP:IO for left and right hippocampus.

5.2.9 Dose Comparison Statistics

Analysis and comparison of the three types of treatment plans was conducted via four techniques. The first was to review the treatment plans and ensure each patient would meet the per-protocol or acceptable deviations criteria of the trial, with reporting indicating either pass or fail. To ensure no bias existed in the treatment planning script, each plan grouping was compared using a two-sided Wilcoxon signed-rank test. Secondly, the dose volume histogram (DVH) for the hippocampi and brain were compared amongst the three plans per patient using a two-sided Kolmogorov-Smirnov test to determine histogram equivalence. Next, Spearman-R correlation coefficients were computed for the dose distribution within the brain between each plan pairing for a given patient. A Fisher-transform was then applied to the correlation coefficients to determine the mean, standard error the mean (SEM) and range. A Friedman's X^2 test was used to determine equivalence among the three plans and a two-sided Wilcoxon sign-ranked test was used to determine equivalence among plan pairings. Using the PyMedPhys python library (version 0.37.1) [120], a 10x resampling gamma analysis [121] was conducted on the low-dose (< 25 Gy) regions within the brain segmentation. Due to the directionality of gamma analysis resampling, the comparison was conducted $\gamma(A \rightarrow B)$, $\gamma(B \rightarrow A)$ with the average of the two gamma values reported. Limiting the analysis to only the hippocampal avoidance regions improved the power of the gamma metric by excluding hot spots, external dose-falloff regions, and the lens avoidance region. For the cohort, the three plans were then compared for equivalence using a Friedman's X^2 test and individual pairings were analyzed with a two-sided Wilcoxon sign-ranked test.

5.3 Results

5.3.1 Contouring

The RTOG 0933 image set provided for this study included 96 patients with both T1 MR and CT images. For the MR images, the median slice thickness and pixel spacing were 1.1 (range 0.6 – 6.0) mm and 0.86 (range 0.39 – 1.09) mm, respectively. The CT images had median slice thickness of 1.375 (range 0.75 – 5.0) mm and pixel spacing of 0.977 (range 0.57 – 2.14) mm. The bilateral hippocampal contours from the trial had median volumes of 4.78 (range 2.06 – 10.2) mL and the single-observer contours had a median volume of 4.03 (range 2.16 – 6.72) mL.

Upon visual inspection of the RTOG hippocampal contours utilizing MIM Software (version 7.1.3), the following data consistency issues were identified: missing slices (n=2), discontinuous volumes (n=1), slices with single voxel contours (n=3). In MIM, protocol compliant contour expansions (5mm) were generated from the RTOG treating physician hippocampi and compared to the provided avoidance volume. Inconsistencies were found between the two expansions, with 7 patients having a disagreement of a HD>3.5 mm. For compliance with the protocol mandated slice thickness, 5 patients had non-compliant CT images (slice thickness > 2.5mm), 3 had non-compliant MR images (>1.5 mm) and 2 had both non-compliant CT and MR images.

Using the trained AG-3D ResNet model instance, predictions were generated from the processed RTOG 0933 images, then each prediction was uncropped and resampled to the original voxel dimensions. The institutional observer (IO) and RTOG treating physician (TP) contours were kept at their original resolutions. Contour correlation was quantified between each contour pairing DL to TP (DL:TP), TP to IO (TP:IO) and DL to IO (DL:IO), with the comparisons given in Table 5.1. For DL and IO contours, an avoidance volume was generated from a uniform 5 mm expansion of the hippocampi and correlation was again computed between the pairings (Table 5.2). Wilcoxon-signed rank test calculated between TP:IO, TP:DL, and IO:DL were statistically significant for 95th, 100th-HD and Dice coefficient for all pairings except TP:IO to DL:IO for right hippocampus Dice (p=0.12).

Visualizations for a series of examples are provided in Figure 5.1, where the HD for TP:IO and TP:DL of each subplot (Figure 5.1, A-E) is indicated in Figure 5.2.A and Figure 5.2.B. Figure 5.1.A represents a TN case where all contour sources agree well. The left hippocampus in Figure 5.1.B depicts an example of a FN prediction for the DL model. Figure 5.1.C depicts a TN case where both the DL and IO vary significantly from the TP contour. Lastly, Figure 5.1.D, Figure 5.1.E show examples of TN prediction for the DL QA, most notably the Figure 5.1.E depicts the only case where the DL model failed to predict a hippocampus, resulting in a TN prediction.

To represent the relative HD performance of TP:DL, and TP:IO, Figure 5.2 (A, B) presents a scatter plot of each for the left and right hippocampus. Figure 5.2 (C, D) provides a vector plot for the change between 100th-percentile and 99th-percentile HD, as well as the ROC curve for both left and right hippocampus. The last row of Figure 5.2 gives a plot Dice coefficient (y-axis) plotted against HD (x-axis) for TP:IO left (Figure 5.2.E) and right (Figure 5.2.F). Indications are provided for cases where the DL model predictions were either TP or FN. Spearman correlation coefficient computed between the Dice coefficient and Hausdorff distance on TP:IO were $\rho = -0.524$ $(p < 0.01)$ and $\rho = -0.366$ $(p < 0.01)$, for left and right hippocampus respectively. A two-sided Mann Whitney U test conducted between the TP and FN groups gave p=0.419 and p=0.031 for left and right hippocampus.

Comparisons were computed between each contouring source, with 100th-percentile HD, 95th-percentile, average surface distance (ASD), relative absolute volume difference (RAVD), Dice, precision, recall and passing rate (HD<7mm) given in Table 5.5. Hausdorff distance (100th and 95th) represent the maximum spatial disagreement between two contours, for which the DL:IO were the closest amongst the cohort and ASD represents the mean spatial disagreement between two contours. From the RAVD, we can see that the DL contours were approximately 15% smaller than the IO contours and 27% smaller than the TP contours, which is supported by precision and recall which penalize over and under prediction. Pass rate and bilateral pass rate are the number of segmentations which agree by HD<7mm, the RTOG inclusion criteria, with the IO:DL agreeing most strongly (70.8% bilateral passing rate). The bilateral functional avoidance volumes were also compared with the same metrics, provided in Table 5.6.

Table 5.5: Contour correlation metrics computed on the RTOG 0933 images (n=96) for the left and right hippocampus structures between the different contour sources. Contours were generated either by institutional observer (IO), deep learning (DL) or by treating physician (TP) during the RTOG trial. Statistics reported as median and inter-quartile range. Bilateral pass indicates a patient which left and right hippocampus pass (HD<7mm) independently.

COMPARISON METRIC	**TP:IO LEFT**	**TP:IO RIGHT**	**TP:DL LEFT**	**TP:DL RIGHT**	**IO:DL LEFT**	**IO:DL RIGHT**
HD (MM)	5.95 (4.79 – 7.75)	5.82 (4.72 – 7.34)	7.23 (5.77 – 9.05)	6.94 (5.39 – 8.71)	4.86 (3.85 – 6.26)	4.74 (3.68 – 6.34)
95TH HD (MM)	2.95 (2.45 – 3.78)	2.65 (2.22 – 3.53)	3.77 (2.97 – 5.49)	3.21 (2.76 – 4.76)	2.40 (1.95 – 2.93)	2.50 (2.00 – 3.13)
ASD (MM)	0.78 (0.60 – 1.07)	0.84 (0.58 – 1.03)	0.90 (0.66 – 1.18)	0.89 (0.69 – 1.12)	0.72 (0.58 – 0.89)	0.74 (0.60 – 1.00)
RAVD (%)	-20.8 (-31.8 - -3.2)	-16.7 (-26.5- -0.6)	-28.9 (-43.3 – -20.0)	-26.6 (-42.1 – -14.2)	-15.6 (-25.0 – 0.20)	-14.6 (-26.1 - -1.2)
DICE	0.69 (0.61 – 0.75)	0.72 (0.62 – 0.77)	0.62 (0.53 – 0.69)	0.65 (0.58 – 0.71)	0.74 (0.66 – 0.78)	0.73 (0.67 – 0.77)
PRECISION	0.79 (0.66 – 0.87)	0.76 (0.67 – 0.85)	0.77 (0.64 – 0.85)	0.77 (0.63 – 0.87)	0.81 (0.71 – 0.86)	0.81 (0.69 – 0.86)
RECALL	0.64 (0.54 – 0.71)	0.67 (0.54 – 0.77)	0.54 (0.43 – 0.61)	0.55 (0.44 – 0.66)	0.69 (0.62 – 0.76)	0.68 (0.59 – 0.76)
PASS RATE (%)	69.8%	69.8%	47.9%	51.0%	81.3%	83.3%
BILATERAL PASS	55.2%		33.3%		70.8%	

Table 5.6: Contour correlation metrics computed on the RTOG 0933 images (n=96) for the hippocampus avoidance structures (5mm expansion of hippocampi). Contours were generated either by institutional observer (IO), deep learning (DL) or treating physicians (TP) in RTOG trial. Statistics are reported as median and inter-quartile range.

COMPARISON METRIC	TP:IO	TP:DL	IO:DL
HD (MM)	7.15 (5.53 – 8.65)	8.18 (6.49 – 9.78)	5.55 (4.50 – 7.12)
95TH HD (MM)	3.32 (2.87 – 4.55)	4.15 (3.38 – 5.69)	2.93 (2.46 – 3.44)
ASD (MM)	1.14 (0.92 – 1.47)	1.29 (1.06 – 1.72)	0.97 (0.84 – 1.15)
RAVD (%)	-18.4 (-25.6- -8.2)	-24.8 (-33.2 - -15.3)	-8.7 (-15.0 – -1.30)
DICE	0.84 (0.80 – 0.87)	0.80 (0.73 – 0.83)	0.87 (0.84 – 0.88)
PRECISION	0.93 (0.89 – 0.96)	0.93 (0.88 – 0.96)	0.91 (0.88 – 0.94)
RECALL	0.76 (0.70 – 0.83)	0.71 (0.62 – 0.77)	0.83 (0.79 – 0.88)
PASS RATE (%)	49.0%	30.2%	70.8%

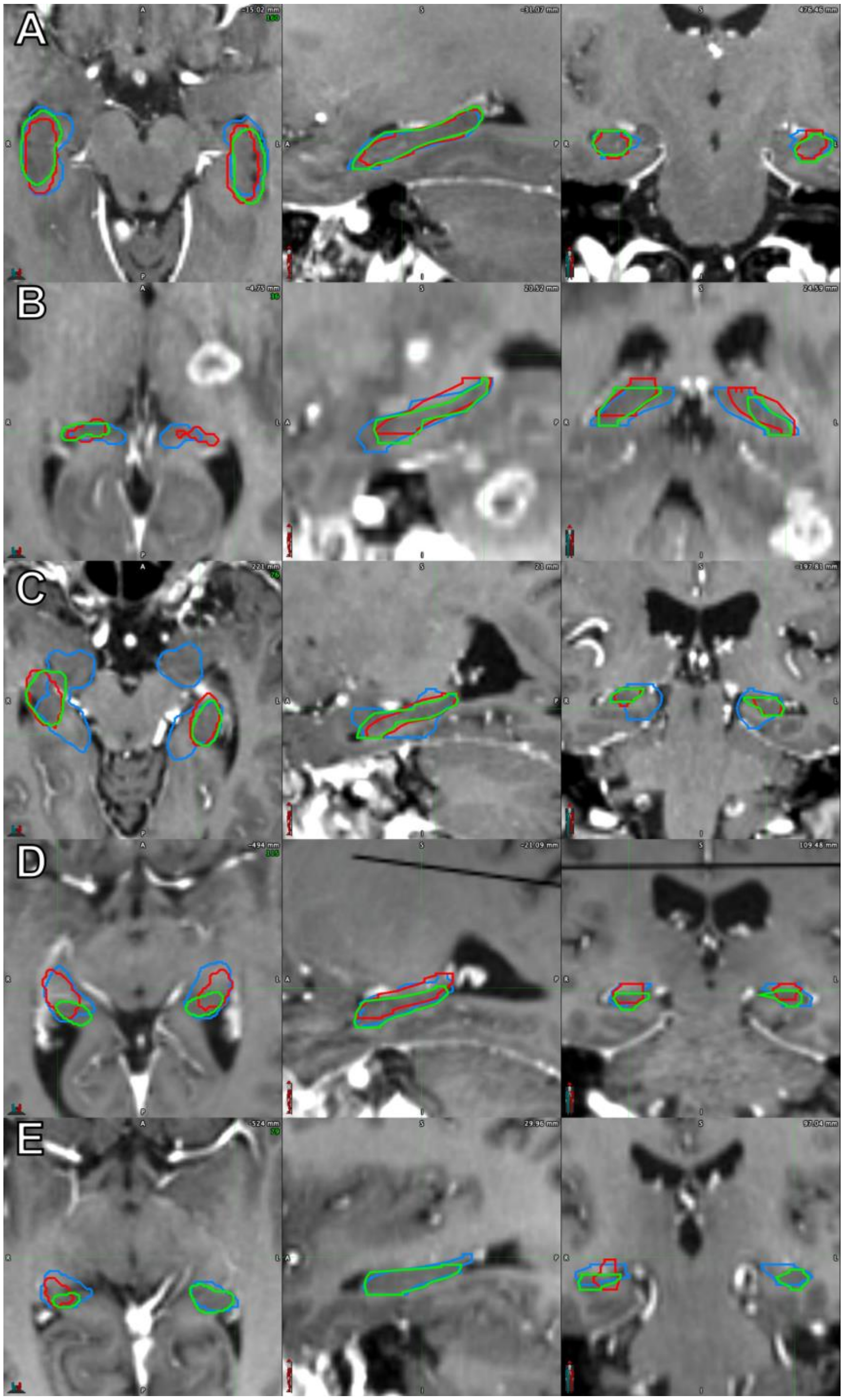


Figure 5.1: Deep learning (red), institutional observer (green), treating physician on RTOG trial (blue) segmentation examples for five cases: all contours agree (HD<7mm) (A), institutional observer contour deviates from treating physician and deep learning (B), treating physician contour deviates from institutional observer (C), deep learning deviates from institutional observer (D), an obvious failure of the deep learning (E). For each example, the percentage occurrence and HD values for each segmentation are provided in Figure 2.

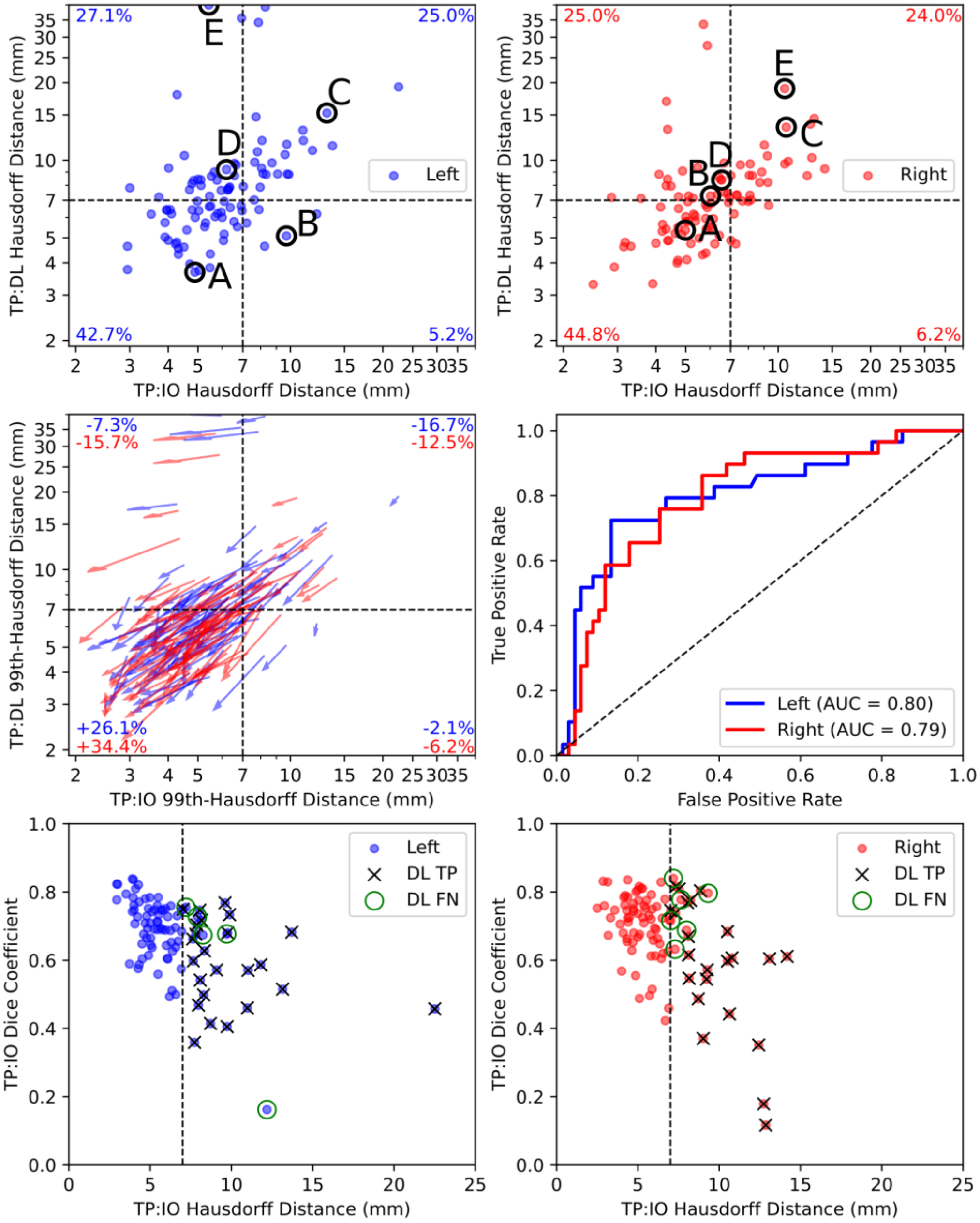


Figure 5.2: TP:IO HD (x-axis) and TP:DL HD (y-axis) for left (A) and right (B) hippocampus, with the percentage of cases per quadrant indicated in each corner and dashed lines at the 7mm threshold. Circles and letters correlate with the segmentations provided in Figure 1. The change in Hausdorff distance from 100th to 99th percentile is given with a vector plot for both hippocampi (C). An ROC AUC is given for the predictive performance of TP:IO failure rate from TP:DL Hausdorff distance (D). Scatter plots of Dice coefficient and Hausdorff distance between TP:IO contours are given for left (E) and right (F) hippocampus. For cases which TP:IO HD>7 mm, indications for if the DL QA determined the case was a true negative (O) or false positive (X) for the RTOG exclusion criterion (HD>7mm).

5.3.2 Dosimetry

Each of the three treatment plans per patient were reviewed by a physician to determine if the plan was per-protocol, or acceptable deviation, for the RTOG-0933 enrollment criteria. The results for that review are given in Table 5.7.

Table 5.7: Physician determined plan adherence to the RTOG-0933 guidelines.

PLAN	PER-PROTOCOL	ACCEPTABLE DEVIATION	UNACCEPTABLE DEVIATION
IO	5	27	0
DL	3	29	0
TP	7	25	0

Dose volume histograms (DVH) for each treatment plan and each contour source pairing were created from data computed using MIM Software. A composite of every patient, plan, and contour permutation of the DVHs is given in Figure 5.3 and descriptive statistics are provided in Table 5.11. The DVHs given are for the hippocampus (blue) and the PTV (red). To quantify the agreement of the DVHs, a two-sided Kolmogorov-Smirnov test was used to determine if the hippocampi DVHs represented a statistically different population (p<0.05). Comparison was conducted between each treatment plan and contour pairing, and the number of hippocampi DVHs which differed statistically significantly are given in Table 5.8.

Table 5.8: Number of DVHs with statistically significant DVH values, defined by Kolmogorov-Smirnov test.

	PLAN IO			PLAN DL			PLAN TP		
CONTOURS	**IO**	**DL**	**TP**	**IO**	**DL**	**TP**	**IO**	**DL**	**TP**
IO	0	23	29	0	32	19	0	3	5
DL	--	0	11	--	0	22	--	0	3
TP	--	--	0	--	--	0	--	--	0

To compute dose correlation across the entire brain treatment volume, a Spearman R correlation coefficient was computed for the dose limited to within the brain region of interest. Friedman's X^2 test was conducted to compare the mean of the three populations (p=0.216), indicating the null hypothesis that the three populations were drawn from the same distribution could not be rejected. For further descriptive data, a Wilcoxon-signed rank test was then used to compute the pairwise per-patient correlation across the two treatment plans. The results of these two tests are provided in Table 5.9 and Table 5.10.

Table 5.9: Mean, standard error of the mean (SEM) and range of the Spearman-R correlation coefficients of the dose within the brain region of interest for DL, IO, and TP treatment plans.

	MEAN	SEM	RANGE
DL:IO	0.370	0.0071	0.127
DL:TP	0.360	0.0071	0.162
TP:IO	0.362	0.0077	0.189

Table 5.10: Wilcoxon signed-rank test between the Spearman correlation coefficients of the dose.

	WILCOXON SIGNED-RANK TEST
DL:IO V. DL:TP	0.0788
DL:IO V. TP:IO	0.295
DL:TP V. TP:IO	0.550

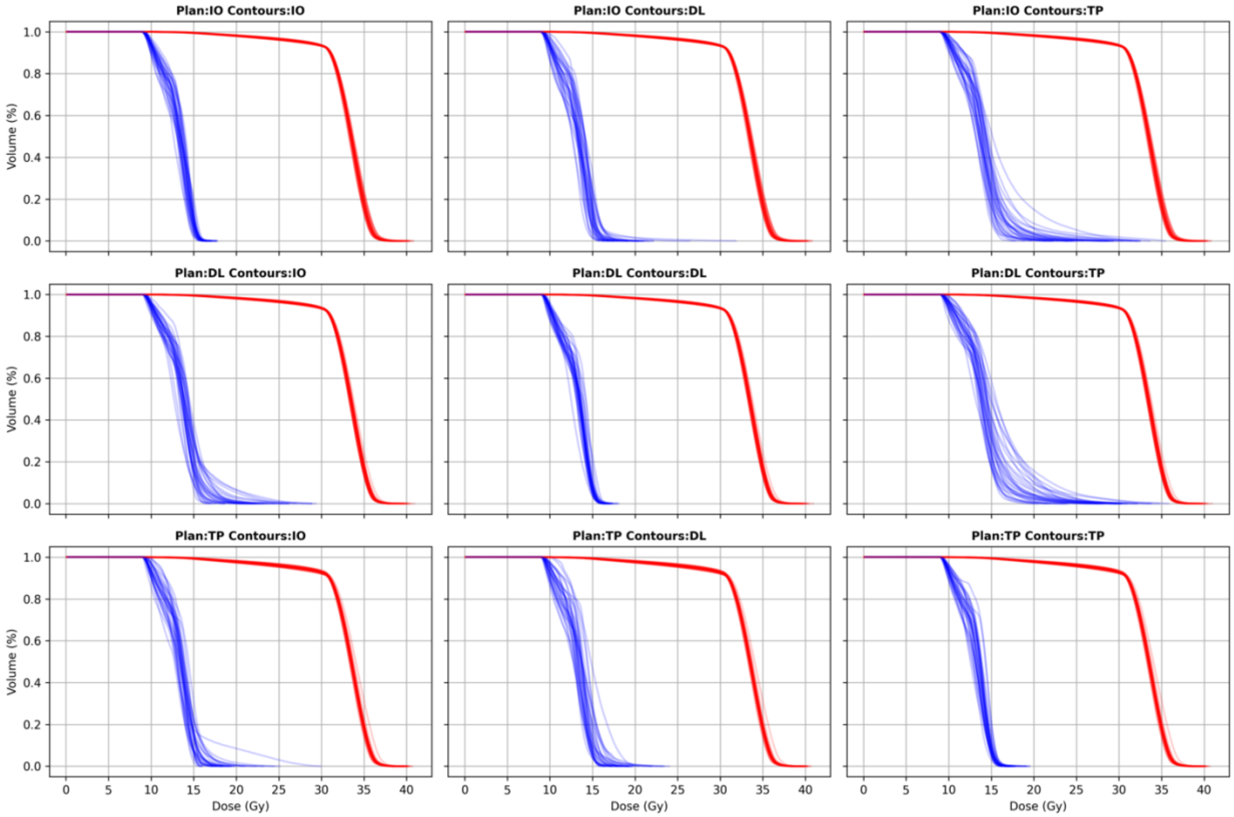


Figure 5.3: Dose volume histograms for hippocampi (blue) and PTV (red) for each of the three contours from each of the three plans. Rows are for each of the different plans (top-to-bottom: IO, DL, TP) and columns are for each contour source (left-to-right: IO, DL, TP). Each plot has all treatment plans (N=32) overlaid one another.

Table 5.11: DVH Metrics (reported in Gy) for the different contour origins and treatment plans. Statistical significance was computed using a Wilcoxon signed-rank test between contour source and treatment plan (across rows). Insignificant (p>0.05) pairings are denoted with superscript numeral.

DVH METRIC	**CONTOUR SOURCE**	**TREATMENT PLAN**		
		IO	**DL**	**TP**
HIPPOCAMPI $D_{100\%}$	IO	9.00 (9.00-9.10)^{1}	9.00 (9.00-9.10)^{1}	9.10 (9.00-9.20)
	DL	9.00 (9.00-9.10)^{2,3}	9.00 (9.00-9.10)^{2}	9.10 (9.00-9.10)^{3}
	TP	9.10 (9.00-9.10)^{4,5}	9.10 (9.00-9.13)^{4}	9.00 (9.00-9.10)^{5}
HIPPOCAMPI D_{MAX}	IO	16.8 (16.5-17.1)	22.3 (20.0-24.8)	18.6 (17.6-19.8)
	DL	18.3 (17.0-20.6)^{6}	17.0 (16.8-17.2)	18.8 (17.4-19.7)^{6}
	TP	25.6 (22.4-29.7)	29.1 (25.9-30.6)	17.3 (17.0-17.8)
PTV $D_{95\%}$	IO	29.9 (29.9-29.9)	29.9 (29.9-30.0)	29.2 (28.9-29.5)
	DL	29.8 (29.7-29.9)	29.9 (29.8-29.9)	29.0 (28.8-29.4)
	TP	30.4 (30.2-30.4)^{7}	30.3 (30.2-30.4)^{7}	29.9 (29.8-29.9)
PTV $D_{98\%}$	IO	24.8 (24.6-25.3)	25.2 (24.7-25.6)	23.4 (23.0-24.4)
	DL	24.6 (24.2-24.9)	25.2 (24.9-25.3)	23.4 (22.7-24.1)
	TP	26.1 (25.6-26.5)	26.4 (25.7-26.7)	25.3 (25.1-25.6)

Using MIM, DVH metrics were generated for each of the trial specific metrics for hippocampus and PTV. When compared using a Wilcoxon signed-rank test, results showed no significant difference in the hippocampus $D_{100\%}$ and the treatment plans for IO and TP generated comparable metrics to the DL contours for hippocampus D_{MAX}. Likewise, the IO and DL plans generated equivalent coverage of the TP PTV with a median $D_{95\%}$ from IO of 30.4 (IQR: 30.2-30.4) and DL of 30.3 (IQR: 30.2-30.4). None of the treatment plans were comparable for $D_{98\%}$, with the TP plan generating significantly worse coverage to the DL and IO contours, while the DL plans yielded the highest coverage of the IO and TP contours.

From each of the treatment plans, the low dose region (< 25 Gy) within the brain was compared using a 3%, 3mm gamma analysis, with median and interquartile range reported (Table 5.12). For demonstration purposes, an example is given in Figure 5.4. Then, the gamma analysis between each treatment plan pairing was compared using a Friedman's X^2 to determine if the three shared an equivalent median (p<0.05) and a Wilcoxon signed-rank test (provided in Table 5.13) to determine the gamma analysis pairings. As this is a test of tests, gamma analysis between two doses, say TP:IO and DL:TP, can be interpreted as a comparison of through an intermediate (DL:IO via TP).

Table 5.12: Gamma analysis of low dose region (<25 Gy) results.

	3% / 3 MM MEDIAN	3% / 3 MM IQR
DL:IO	0.669	0.620 – 0.730
DL:TP	0.670	0.606 – 0.768
IO:TP	0.722	0.647 – 0.755

Table 5.13: Wilcoxon signed-rank test to compare the gamma analysis scores.

	WILCOXON SIGNED-RANK TEST
DL:IO V. DL:TP	0.489
DL:IO V. TP:IO	0.0017
DL:TP V. TP:IO	0.0315

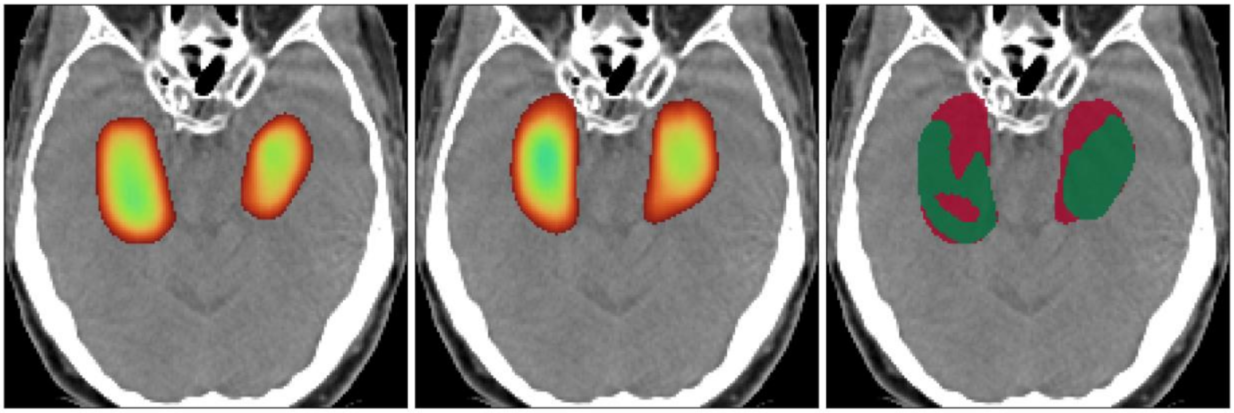


Figure 5.4: A 3%, 3mm Gamma analysis (right) displayed with per-voxel passing (green) and failing (red) for the combined dose region of both plans. The given gamma analysis was computed between less than 25Gy regions within the brain contour of the DL treatment plan dose (left) and IO treatment plan dose (middle).

5.4 Discussion

5.4.1 Contour Discussion

In this investigation we have demonstrated the feasibility of training a DL model on a single institution dataset for the application of hippocampal contour QA on a multi-institutional trial. As evidenced by the significantly higher performance of DL:IO over DL:TP across multiple metrics, deep neural networks are capable of strongly replicating the contouring style of the training institution. This enables a DL model to function as a high-sensitivity, first-pass QA tool. A DL QA tool can also be published alongside the clinical trial results, ensuring the new radiotherapy paradigm is implemented in an equivalent manner. While the effort required to assemble a training dataset is non-trivial, the 390 patients used in this study is not indicative of the required number to saturate the AG-3D ResNet, and future investigations are needed to develop a specific cohort size. Alternatively, a multi-staged approach to training may be used where a model is trained from a smaller set and used to generate predictions on a larger dataset, from which the contours are edited and used to re-train a final instance.

Although the DL model replicated the contouring style of the training institution, the predicted contours were of smaller relative absolute volume difference (RAVD) than the ground truth. The smaller contours could be attributable to the non-functional mapping of the ground truth or from using a Dice loss function. In our training dataset, each image corresponded to up to three unique contours, creating the possibility that the DL model resorted to learning the intersection of the three contours. If that were the case, we would expect training on the consensus contour would result in larger predictions. An alternative explanation is that the Dice loss function incentivizes smoother, more certain predictions and could be addressed with a combined Dice-Focal loss or rind Dice loss function.

While we used the RTOG protocol acceptability criteria of HD<7mm, it is evident from the Spearman correlation coefficients computed between TP:IO (Figure 5.2.E, Figure 5.2.F) that Hausdorff distance and Dice correlation coefficient are weakly correlated on volumetrically small structures. The 100th-percentile Hausdorff distance is extremely sensitive to treatment planning system contour smoothing, expansion algorithms and image voxel sizes. This leads to high Dice coefficient contours which fail by a small margin, or low Dice contours which barely pass.

Furthermore, the segmentations designated as false negatives by the deep learning quality assurance were found to be significantly different (p=0.419 and p=0.031 for left and right hippocampus) when compared with a two-sided Mann Whitney U test, a non-parametric test of equivalence for un-paired data. The left hippocampus populations would have been significantly different if the outlier had been excluded. These results indicate that despite failing the RTOG inclusion criteria, the false negative predictions yielded large Dice correlation coefficients than the true negative predictions.

The left hippocampus false negative outlier, within Figure 5.2.E, with low dice performance can be seen where the DL model failed to identify IO:TP disagreement. For this case, the T1-MRI provided had voxel dimensions of $0.9 \times 0.9 \times 6.0$ mm and a CT of $1 \times 1 \times 1$ mm. Generating contours on a thick slice MRI exacerbates the differences in the segmentation interpolation functions, and the contours created from MIM Software’s interpolation exhibited less smoothness than the RTOG treating physician contours. Alternatively, the treating physician may have generated contours on a higher-resolution sequence which was not provided for this study. While the DL:TP prediction for this case passed the RTOG criterion (HD=6.19mm) for left hippocampus, the Dice correlation coefficient was very poor (Dice=0.22). This highlights the relative insensitivity of HD for evaluating correlation and questions its utility as a stand-alone metric for segmentation comparison. Furthermore, this case exhibits the strength of a CT-only first-pass QA tool in that contours on the CT frame of reference are the ground truth. Thereby, predicting from CT-only eliminates uncertainty in contour interpolation and image registration seen across multiple institutions.

The reliance on traditional correlation metrics for contemporary clinical trial QA places the burden on trial designers to select a robust and clinically correlated metric prior to enrollment.

This design process is akin to expert system feature engineering and faces many of the shortcomings we've seen deep learning address in recent years. While we have shown deep learning is capable of segmentation, the applicability of a QA tool is limited by this reliance on traditional metrics such as Dice and HD. In future work we intend to explore the application of deep learning models for the quantification of contour correlation to forego the need for classical thresholds.

5.4.2 Dosimetry Discussion

From the DVHs provided in Figure 5.3, as expected, the hippocampus dose is lowest for matching contour-plan pairs, with the TP-TP plan-contour pair having the largest variance in hippocampal dose and the IO-IO plan-contour pairing having the lowest. When we compare the non-matching pairs, it becomes clear that the volumetrically larger TP contours lend themselves to lower hippocampal doses on the IO and DL contours because the avoidance regions are generally larger. While the avoidance regions allow for more sparing, that is with the trade-off of reduced prescription coverage, which is not easily displayed in a DVH for a large PTV. Interestingly, the IO-TP and DL-TP pairings appear to be comparable in their hippocampal dose spreads, with both larger than the DL-IO hippocampal DVH spread. These DVH comparisons ultimately raise the question of the clinical goals for the patient treatment and the balance between coverage and neurocognitive function. For the future roll-out of a clinical tool for hippocampal segmentation, a model trained to bias volumetrically larger contours may be favored to generate plans comparable to those from the RTOG-0933. In instances where the eventual end-user treating physician is more concerned about PTV coverage than hippocampal avoidance, an alternative model with volumetrically smaller contours would allow physicians to tailor plans more well-suited to a patient's specific clinical needs.

5.5 Conclusion

The work presented in this chapter demonstrates the feasibility of using a single-intuitional, non-HA-WBRT dataset to train a CT-only deep learning neural network to use as a first-pass hippocampal contour QA tool on a multi-institutional dataset.

CHAPTER 6 Techniques Translated to 4DCT-Perfusion

6.1 Introduction

Technetium-99m (^{99m}Tc) labeled macroaggregates of albumin (MAA) imaged with single photon emission computed tomography (SPECT) is considered the standard method for the quantitative determination of pulmonary perfusion [122]. Spatially correlated x-ray computed tomography (CT) images allow for attenuation and scatter correction of SPECT [123,124], while also evaluating lung anatomy with accuracy rivaling CT angiography [125]. While static CT images contain only anatomic information, dynamic (4DCT) images also contain functional information [126]. For example, 4DCT Ventilation Imaging (4DCT-VI) is a technique to derive ventilation images from the inhale and exhale 4DCT phases [127,128]. An intrinsic convenience of this approach in oncology patients is that these images are extracted from routinely acquired treatment planning 4DCTs. 4DCT-VI and its use in radiation therapy (RT) treatment planning to preserve pulmonary function [129,130] has been an active area of research and the subject of prospective clinical trials [131].

Over the past three decades ^{99m}Tc-MAA perfusion imaging has been the focus of research to understand and reduce pulmonary injury following thoracic radiotherapy. At 2-4 months post-radiotherapy, pulmonary perfusion was found to decrease, with respect to pre-treatment values, in proportion to the local radiation dose [132]. The dose dependent perfusion loss could be detected with SPECT imaging for 12 months following treatment [133]. Lee *et al.* [134] performed a meta-analysis on radiotherapy planning studies which sought to use SPECT or PET functional lung imaging to avoid dose to the functional lung. Delivery of radiotherapy through hypo-perfused pulmonary regions for lung cancer treatment was shown to result in less pulmonary injury in a prospective trial [135]. Additionally, a retrospective evaluation of two prospective studies found functional lung avoidance planning may promote increased post-treatment perfusion in low-dose regions for select patients [136]. Physiological images used in radiotherapy treatment planning for image guidance to avoid the irradiation of highly functional regions remains an active area of research [137,138]. However, pulmonary functional imaging based on ^{99m}Tc-MAA SPECT is not broadly available for treatment planning in radiation oncology clinics.

Reports indicate that respiration-induced lung blood mass variations are measurable on 4DCT [127] and match the expected physiology changes [139]. This finding was further characterized in a retrospective study of 89 patients who received 4DCT for radiation therapy treatment planning [140], which found a relationship between pulmonary tidal volume and changes in pulmonary blood mass during respiration. Attempts to make 4DCT perfusion images calculated on a voxel-by-voxel basis resulted in unreliable measurements [141] as the signal is overwhelmed by random noise. Additionally, an exact relationship between the respiratory-induced pulmonary blood mass dynamics observed on 4DCT and pulmonary perfusion is unknown, complicating the production of pulmonary perfusion images from 4DCT. Deep learning allows computational models to identify intricate structures within a data set without an *a priori* knowledge of the relationship [49]. Therefore, deep learning has the potential to detect the pulmonary perfusion signal from 4DCT alone.

Prior work by Zhong *et al.* [142] utilized a deep learning network to generate synthetic ventilation images. As their ground truth, Zhong *et al.* used a synthetic image which was computationally derived from 4DCT using an image registration technique [143]. Jang *et al.* [144] generated SPECT perfusion from CT images with a 2D-UNet [57] based cGAN [145] model trained on the inhalation CT acquired during the SPECT/CT scan. As the authors note, the 2D model design presented inherent performance limitations, notably making the predicted volumes prone to discontinuities. Ren *et al.* [146] utilized a 3D-UNet [60] to generate 11-bin discretized synthetic

MAA-SPECT perfusion imaging from a single free breathing CT volume. Ren *et al.* [147] continued their investigation of generating discretized synthetic perfusion images with a modified model design and an expanded cohort with 12 out of 73 patients having lung cancer.

We propose a convolutional neural network model to generate synthetic pulmonary perfusion images from 4DCT alone. Our technique utilizes a 3D deep learning model, is trained from clinically acquired data and forgoes manual lung segmentation prior to prediction. The resulting images to ^{99m}Tc-MAA SPECT-CT perfusion images along with derived F50 functional avoidance contours [148] derived from the synthetic images.

6.2 Methods

6.2.1 Clinical Data Acquisition

A retrospective data set was compiled from imaging studies collected in an ancillary study to a prospective clinical trial [149] (NCT02528942). The ancillary study collected pre- and post-RT SPECT/CT perfusion and pre- and post-RT 4DCT images on the institutional subset of patients on the clinical trial. Average temporal separation between SPECT and 4DCT was 6 days (IQR = 3.5 days) and pre- to post-treatment was 154 days (IQR = 12.5 days). The 4DCT images (120 kVp, 3mm axial thickness) were acquired in a supine hands-over-head position during normal tidal breathing with a flat-couch 16-slice Philips Brilliance Big Bore CT scanner (Philips Healthcare, Andover, MA) utilizing the oversampled spiral 4DCT acquisition technique [150]. Following administration of 4.0 mCi of ^{99m}Tc-MAA (Lantheus Medical Imaging, Billerica, MA), perfusion images were acquired in a supine hands-over-head position during tidal respiration using a curved-couch dual head Siemens Symbia SPECT and 2-slice CT-scanner (Siemens Medical Solutions, Malvern, PA). The scanner was configured with a high-resolution parallel hole collimator, a 15% energy window with centerline at 140 keV and CT at 130 kVp and 50 - 100 mAs (weight dependent). 3D attenuation corrected SPECT images were reconstructed transaxially using an iterative ordered subset expectation maximization (OSEM) algorithm and post-processed with a 5mm gaussian blur. SPECT volumes were constructed with axial dimensions of 64×64 (7.8125×7.8125 mm voxels) and 1.5 mm trans-axial slice thickness. All images were stored in the hospital PACS in DICOM format.

6.2.2 Data Preparation

Images were exported to a MIM workstation (MIM Software, Cleveland, OH) and to represent the typical planning image volume, an average intensity projection CT image (AIP-CT) was generated for each 4DCT. Using the MIM rigid image registration algorithm, the CT image acquired during the SPECT scan was registered to the AIP-CT. The rigid registration algorithm was used instead of the deformable registration algorithm due to non-physiological shear in the deformable vector field in low-contrast regions, as evidenced by the deformable vector field curl (Appendix Figure C.1) [151]. The resultant displacement vector field was used to register the SPECT to the AIP-CT image. For later statistical analysis, bilateral lung contours (excluding trachea and bronchi) were taken from the AIP-CT clinical treatment planning contours or manually created. Using python (version 3.8.5), each DICOM image was reconstructed into a NumPy array with our open source DICOM management package [113]. On the AIP-CT, a body contour was generated from voxels greater than 25% of the maximum CT value. Using the body contour center of mass as the centroid, the images were cropped to $280 \times 280 \times 110$ voxels. For image volumes with less than 110 axial slices, the superior border of the volume was padded with zeros. Then each image was down-sampled (2:1) with a mean value function, yielding image dimensions $140 \times 140 \times 55$ voxels (approximately $330 \times 330 \times 330$mm). Resampling the inputs reduced the computational demands while maintaining resolution ($2.35 \times 2.35 \times 6$ mm) finer than the configured SPECT imaging system (7.5 mm) [152]. Finally, each cropped inhale-exhale CT pair and SPECT volume was normalized.

6.2.3 Model Design and Training

The processed data was used to train a model based upon the High-Res3DNet [61] (ResNet), implemented in Tensorflow version 2.3 [153]. The model (Figure 6.1) is a residual network consisting of three groups of increasing dilated convolutional operations (dilation of one, two and four). Within the network, each dilation level is repeated in three residual blocks. Each convolution operation has a kernel of size $3 \times 3 \times 3 \times N$ (N of 16, 32, 64 for each dilation level), with a rectified linear unit activation. In total, the network is comprised of 813,297 trainable parameters. The foundational model design was modified with the addition of dropout layers in each residual block and a sigmoid final activation. This model was trained to predict the SPECT perfusion image from the 0% and 50% phases of the 4DCT. The efficiency of this network has been explored in our prior publication on another CT-based imaging task [111]. Furthermore, this model architecture has been demonstrated in other publications with state-of-the-art results in synthetic imaging tasks [154,155].

Table 6.1: Tuned hyperparameter values used for five-fold cross-validation.

HYPERPARAMETER	VALUE
LEARNING RATE	1.25E-3
LEARNING RATE REDUCTION	0.5x after plateau of 3 epochs
DROPOUT RATE	0.25
AUGMENTATION	±10° axial rotation ±10% xy-axis shift
ASYMMETRY FACTOR	1.35
EPOCHS	40 epochs of 150 steps
BATCH SIZE	4

Mean squared error (MSE) and mean absolute error (MAE) are common loss functions for regression due to their simplicity and symmetry to over- and under-predictions. For tasks which require increased penalization of under-prediction, relative to over-prediction, a logarithmic activation to MSE can be applied. For the accurate prediction of perfusion defects, we surmise that a loss function with greater penalization for over-prediction of the ground truth is required to prevent the model from uniformly predicting healthy perfusion and incentivize the proper prediction of hypo-perfused lung regions. To achieve this, we devised a tunable, asymmetrical loss function scaling factor (Equation 6.1) to alter the ratio of penalization for over-prediction relative to under-prediction. In Equation 6.1, f represents a regression loss function (e.g., MAE, MSE), scaled by an asymmetrical factor dependent on the difference (Δ) between the normalized ground truth and prediction. The asymmetrical factor is tuned via a positive scalar value (α). The magnitude of the asymmetrical scalar (β) is defined by Equation 6.2. When $\alpha \rightarrow \infty, \beta \rightarrow 1$ and conversely, $\alpha \rightarrow 0, \beta \rightarrow \infty$, dictating the relative penalization of false positive predictions. For training our model, we paired this asymmetrical scaling factor with the MAE loss function, yielding an asymmetrical mean absolute error (AMAE) loss function.

$$f' = f * \frac{\log(2)}{\log(2+\Delta+\alpha)} \tag{6.1}$$

$$\beta = \frac{f\prime(-\Delta)}{f\prime(\Delta)} = \frac{\log(3+\alpha)}{\log(1+\alpha)} \tag{6.2}$$

Hyperparameter tuning was conducted on a subset of the validation set using a manual grid search. Final hyperparameters are reported in Table 6.1. Model training was parallelized across two Nvidia Quadro RTX8000 GPUs (Nvidia, Santa Clara, CA). A five-fold cross-validation, split by patient, was conducted for twenty instances per fold while holding hyperparameters constant. Inference required 0.45 seconds per volume when conducted on the Nvidia Quadro RTX8000.

6.2.4 Evaluation Metrics

A cross-fold validation was conducted for trained model instance selection prior to evaluation on a hold-out test set. Pearson and Spearman correlation coefficients were calculated between the clinical and predicted perfusion images. Masked array type Numpy objects, created with manual lung segmentations, were used to limit the correlation coefficients to the lung voxels, thereby negating correlation biases (e.g., volumetric, background). Spearman and Pearson correlation coefficients were calculated with 20 iterative predictions (each from a sequentially trained, randomly initialized model). Standard error of the mean (SEM) and range of the 20 iterative predictions was obtained for each image. Correlation metrics for the cohort were determined to be non-normally distributed by the Kolmogorov-Smirnov test. Therefore, the correlation coefficients were normalized using Fisher's z-transformation. Next, the normalized coefficients were averaged for all cases. These average Fisher's z-transformed coefficients were then converted back to correlation coefficients and reported with mean (standard deviation) and median (interquartile) summary statistics. An overall correlation coefficient, and 95% confidence intervals, was estimated using both fixed and random effects modeling for the three subsets (all, pre-, post-RT). For patients with pre- and post-RT imaging studies, we compared median coefficients using the non-parametric Wilcoxon signed-rank test. Statistical significance was accepted at $P < 0.05$, and all statistical analyses were performed as two-sided tests. All statistics were completed using R (R version 4.0.0 (2020-04-24)).

For each pre-RT study (N=32), perfusion functional avoidance contours were generated with the 50th percentile binary threshold (F50) technique [148]. The contours were cleaned with two morphological operations (binary closing and fill holes) to reduce any pinholes in the contours which may be problematic for treatment planning. The clinical and synthetic F50 segmentations were compared with Dice coefficient, average surface distance (ASD), 95th-percentile Hausdorff distance (HD95), relative absolute volume difference (RAVD), precision, recall and F-score. The per patient mean of the 20 iterative predictions was computed and a median and interquartile range was reported for the cross-fold validation population. Contour comparison metrics were computed using python (version 3.8.5) with the MedPy library [118].

From each of five folds, a top performing model was identified based upon Spearman correlation coefficient on the validation set. On a hold-out test set of five patients (eight studies) predictions were generated for each of the five models and the correlation coefficient, statistical analysis, forest plots and contour generation were repeated for those predictions. Final predictions, as would be used in a clinical setting, were created using two methods: majority voting amongst the five model instances and a single prediction from the model instance with the highest validation performance across all models. For both inference methods Spearman and Pearson correlation coefficients were calculated and F50 functional contour agreement was quantified.

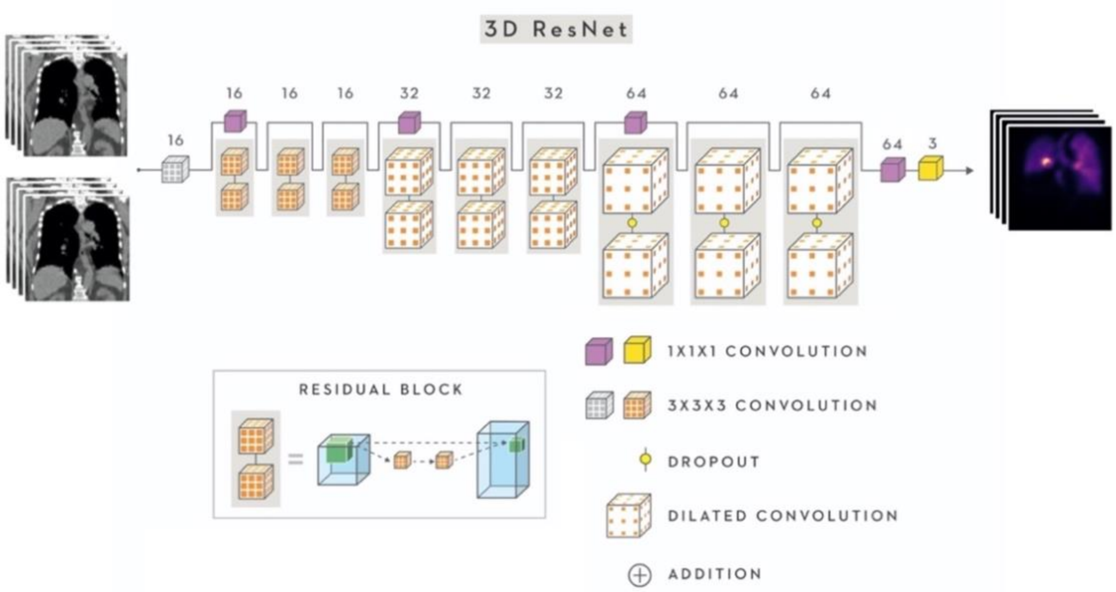


Figure 6.1: A representation of the AG-3D ResNet model, with the inhale and exhale volumes as input and the predicted SPECT volumes as output.

6.3 Results

Spearman correlation coefficients computed from the cross-fold validation on the pre- and post-RT studies are given (Figure 6.2, Figure 6.3) and Pearson correlation coefficients are provided in the appendix (Appendix Figure C.2, Appendix Figure C.3). Comparison of the pre- and post-

RT populations shows no significant performance difference for either Spearman (p=0.19) or Pearson (p=0.29) correlation coefficients (Appendix Figure C.5). This indicates the model, on the validation set, does not uniformly predict the expected healthy lung behavior for all studies. As a visual representation of the population mean of the Spearman correlation for the cross-fold validation (0.730) a study with a nearly equal correlation (0.731) is given in Figure 6.4. A further spectrum of correlation coefficients computed on the individual 2D coronal slices (Appendix Figure C.2) is provided in the supplement to assist in visualizing the range of observed Spearman correlation coefficients. For each pre-RT validation case (N=24), the F50 functional avoidance contours were computed, and the comparative statistics are reported in Appendix Table C.1.

From each cross-fold (N=5), the top performing model instance was selected by Spearman correlation on the cross-fold validation. Forest plots of Spearman (Figure 6.5) and Pearson (Figure 6.6) correlation coefficients generated a fixed effects model estimate for Spearman correlation of 0.63 (95% CI: 0.25-0.84) and Pearson correlation of 0.70 (95% CI: 0.36-0.88). Final predictions using both the single model and majority vote techniques are provided in Table 6.2. Visualizations of the majority vote predictions on the test set studies with the lowest performance (Figure 6.7) and highest performance (Figure 6.8) are also given. For both inference techniques, functional avoidance contours comprised of the well perfused lung was generated from the pre-RT clinical and synthetic perfusion images. The contour comparative statistics for the single model statistics are given in Table 6.3 and the majority vote technique is given in Table 6.4. The synthetic and clinical derived functional contours were found to correlate well, with the majority vote technique yielding results with a Dice score of 0.803 (IQR: 0.750 – 0.810) and average surface distance of 5.92mm (IQR: 5.68 – 7.55).

Table 6.2: Performance of a single model inference and a majority voting for each study in the hold-out test set, with metrics calculated using the lung-mask technique.

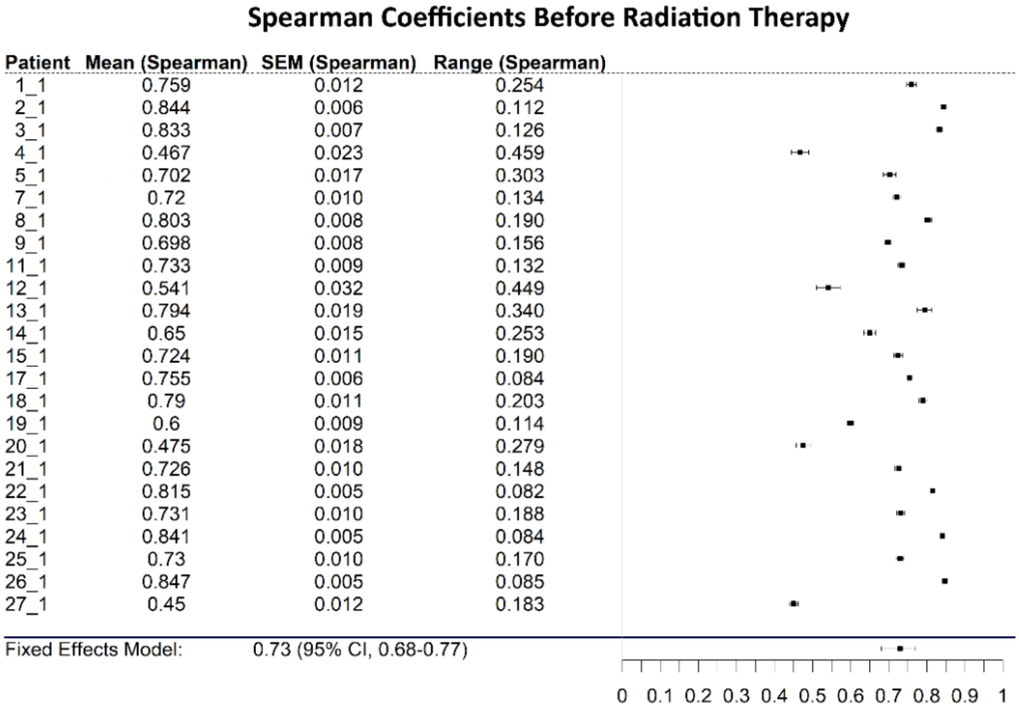
	SINGLE MODEL		MAJORITY VOTE	
STUDY	PEARSON	SPEARMAN	PEARSON	SPEARMAN
28_1	0.609	0.634	0.638	0.655
29_1	0.681	0.712	0.730	0.753
29_2	0.555	0.664	0.494	0.643
30_1	0.639	0.677	0.753	0.777
31_1	0.758	0.787	0.727	0.760
31_2	0.661	0.739	0.685	0.764
32_1	0.247	0.312	0.294	0.359
32_2	0.473	0.517	0.475	0.513

Table 6.3: Single model functional avoidance contour agreement statistics.

STUDY	DICE	ASD (MM)	HD95% (MM)	PRECISION	RECALL	F-SCORE
28_1	0.749	5.851	20.73	0.749	0.748	0.749
29_1	0.798	6.333	18.00	0.793	0.804	0.798
30_1	0.760	5.023	15.41	0.762	0.759	0.760
31_1	0.839	5.828	15.23	0.839	0.839	0.839
32_1	0.629	8.032	25.67	0.629	0.635	0.629

Table 6.4: Majority vote functional avoidance contour agreement statistics.

STUDY	DICE	ASD (MM)	HD95% (MM)	PRECISION	RECALL	F-SCORE
28_1	0.750	5.683	23.43	0.751	0.748	0.750
29_1	0.810	5.918	16.78	0.811	0.809	0.810
30_1	0.803	3.850	11.37	0.804	0.801	0.803
31_1	0.825	8.028	19.22	0.826	0.824	0.825
32_1	0.639	7.552	26.10	0.641	0.638	0.639

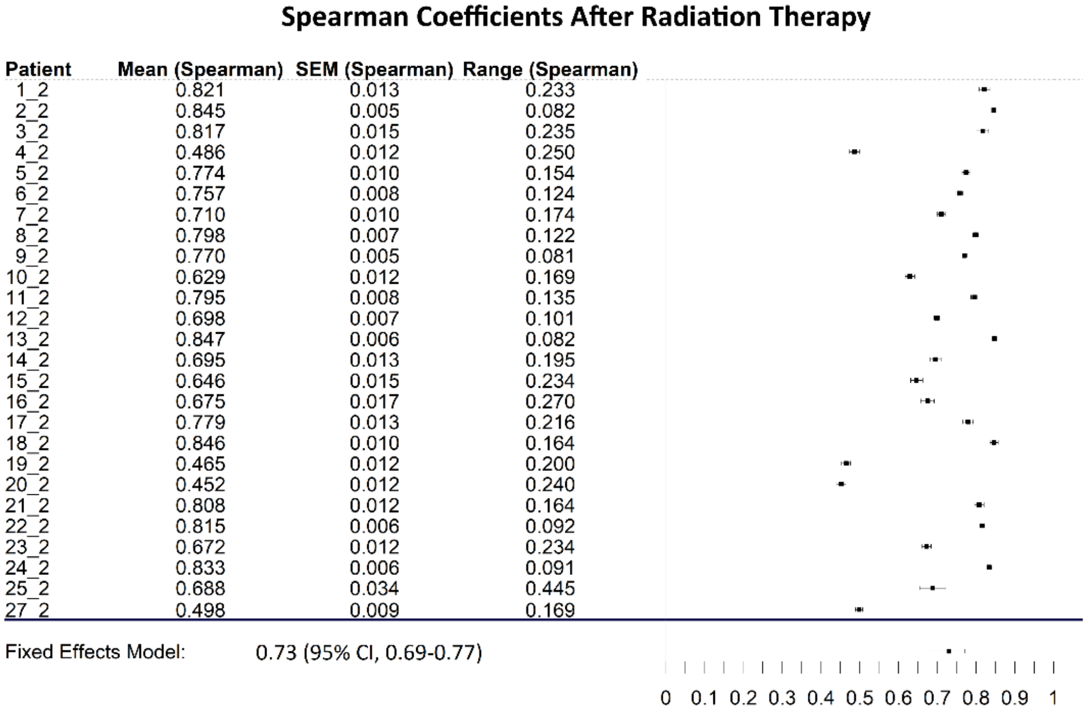


Spearman Coefficients Before Radiation Therapy

Patient	Mean (Spearman)	SEM (Spearman)	Range (Spearman)
1_1	0.759	0.012	0.254
2_1	0.844	0.006	0.112
3_1	0.833	0.007	0.126
4_1	0.467	0.023	0.459
5_1	0.702	0.017	0.303
7_1	0.72	0.010	0.134
8_1	0.803	0.008	0.190
9_1	0.698	0.008	0.156
11_1	0.733	0.009	0.132
12_1	0.541	0.032	0.449
13_1	0.794	0.019	0.340
14_1	0.65	0.015	0.253
15_1	0.724	0.011	0.190
17_1	0.755	0.006	0.084
18_1	0.79	0.011	0.203
19_1	0.6	0.009	0.114
20_1	0.475	0.018	0.279
21_1	0.726	0.010	0.148
22_1	0.815	0.005	0.082
23_1	0.731	0.010	0.188
24_1	0.841	0.005	0.084
25_1	0.73	0.010	0.170
26_1	0.847	0.005	0.085
27_1	0.45	0.012	0.183

Fixed Effects Model: 0.73 (95% CI, 0.68-0.77)

Figure 6.2: Forest plot of the validation set, pre-treatment Spearman correlation coefficients. Encoding of the study is by patient with '_1' representing a pre-treatment imaging study.



Spearman Coefficients After Radiation Therapy

Patient	Mean (Spearman)	SEM (Spearman)	Range (Spearman)
1_2	0.821	0.013	0.233
2_2	0.845	0.005	0.082
3_2	0.817	0.015	0.235
4_2	0.486	0.012	0.250
5_2	0.774	0.010	0.154
6_2	0.757	0.008	0.124
7_2	0.710	0.010	0.174
8_2	0.798	0.007	0.122
9_2	0.770	0.005	0.081
10_2	0.629	0.012	0.169
11_2	0.795	0.008	0.135
12_2	0.698	0.007	0.101
13_2	0.847	0.006	0.082
14_2	0.695	0.013	0.195
15_2	0.646	0.015	0.234
16_2	0.675	0.017	0.270
17_2	0.779	0.013	0.216
18_2	0.846	0.010	0.164
19_2	0.465	0.012	0.200
20_2	0.452	0.012	0.240
21_2	0.808	0.012	0.164
22_2	0.815	0.006	0.092
23_2	0.672	0.012	0.234
24_2	0.833	0.006	0.091
25_2	0.688	0.034	0.445
27_2	0.498	0.009	0.169

Fixed Effects Model: 0.73 (95% CI, 0.69-0.77)

Figure 6.3: Forest plot of the validation set, post-treatment Spearman correlation coefficients. Encoding of the study is by patient with '_2' representing a post-treatment imaging study.

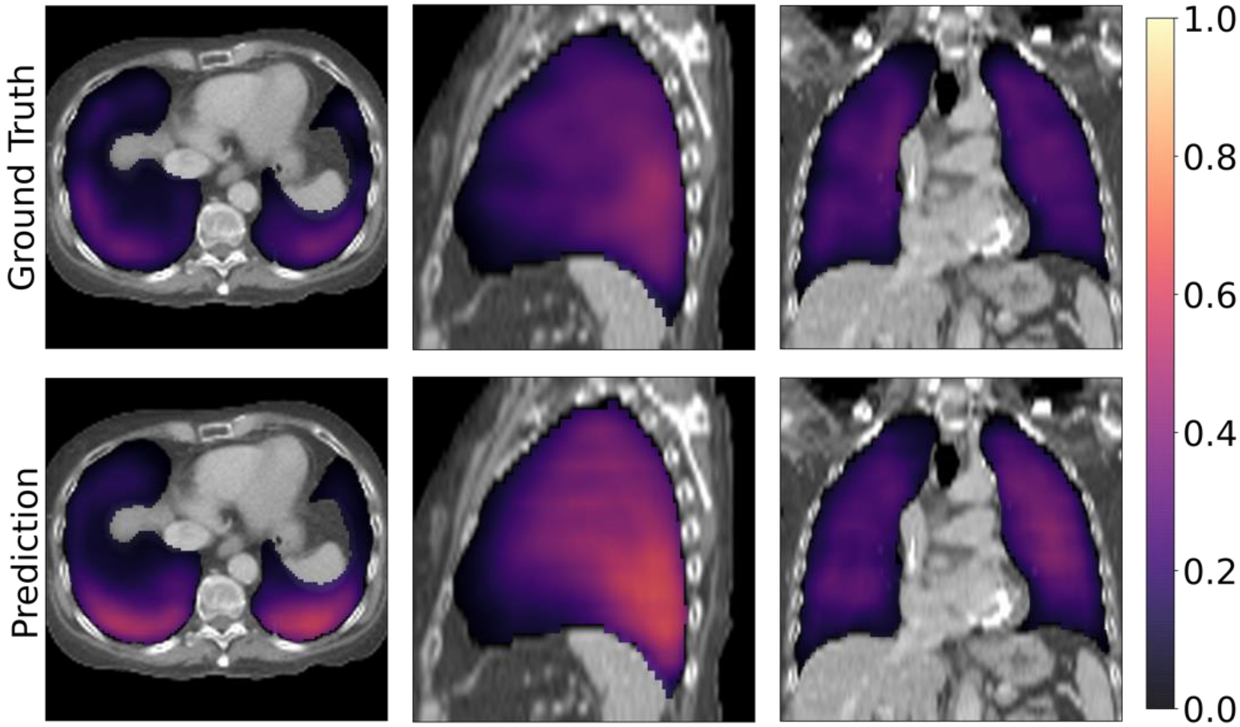
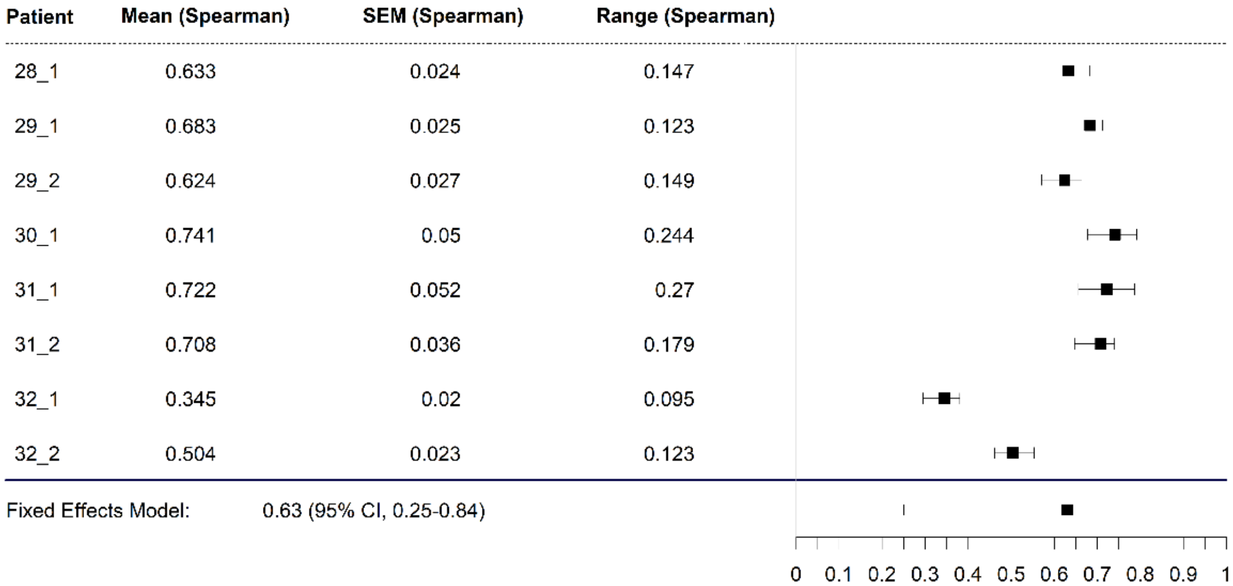


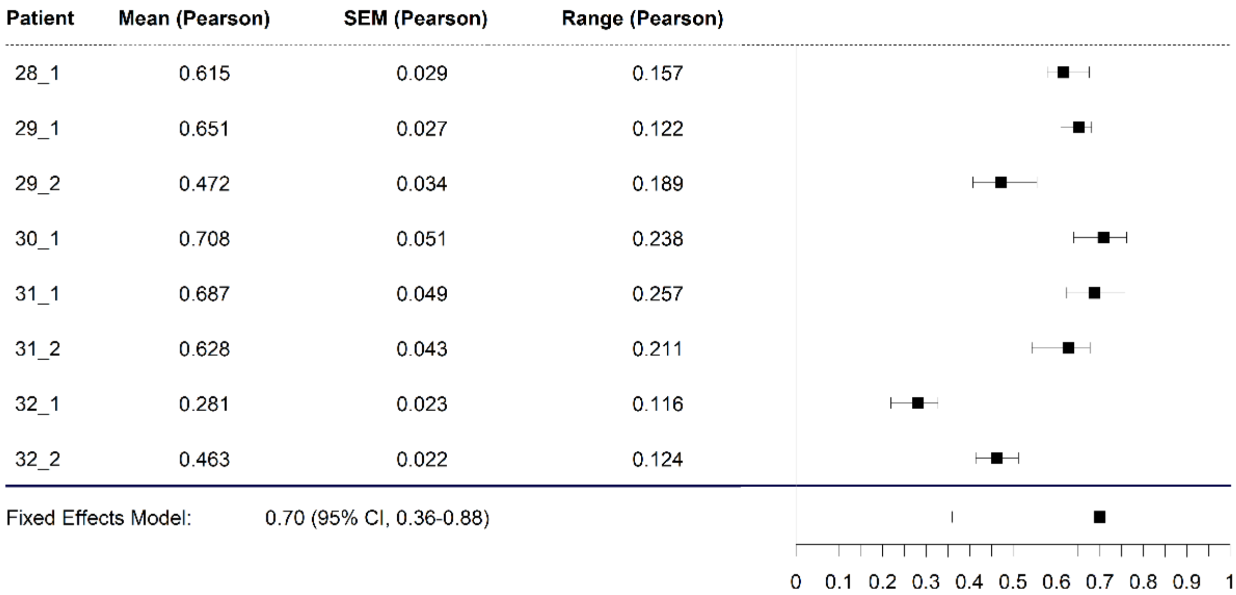
Figure 6.4: A visual comparison of the ground truth (above) and prediction (below) for study 23_1 (Spearman 0.731; SEM: 0.01), chosen as a representation of the population mean of the cross-validation (Spearman: 0.73; 95% CI: 0.68-0.77).



Patient	Mean (Spearman)	SEM (Spearman)	Range (Spearman)
28_1	0.633	0.024	0.147
29_1	0.683	0.025	0.123
29_2	0.624	0.027	0.149
30_1	0.741	0.05	0.244
31_1	0.722	0.052	0.27
31_2	0.708	0.036	0.179
32_1	0.345	0.02	0.095
32_2	0.504	0.023	0.123

Fixed Effects Model: 0.63 (95% CI, 0.25-0.84)

Figure 6.5: Forest plot of Spearman correlation coefficients for test set imaging studies. Encoding of the study is by patient followed by a number indicating pre-treatment (1) or post-treatment (2).



Patient	Mean (Pearson)	SEM (Pearson)	Range (Pearson)
28_1	0.615	0.029	0.157
29_1	0.651	0.027	0.122
29_2	0.472	0.034	0.189
30_1	0.708	0.051	0.238
31_1	0.687	0.049	0.257
31_2	0.628	0.043	0.211
32_1	0.281	0.023	0.116
32_2	0.463	0.022	0.124

Fixed Effects Model: 0.70 (95% CI, 0.36-0.88)

Figure 6.6: Forest plots of Pearson correlation coefficients for test set imaging studies. Encoding of the study is by patient followed by a number representing pre-treatment (1) or post-treatment (2).

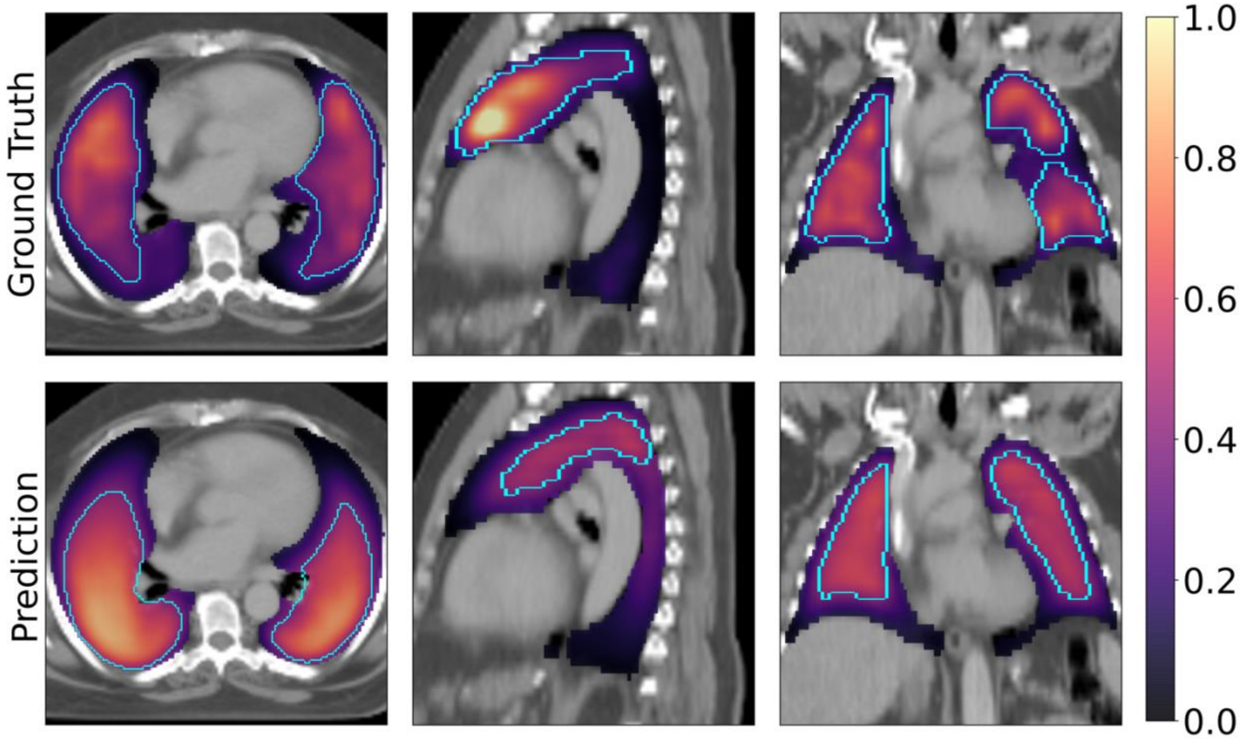


Figure 6.7: The test set study (32_1) with the lowest performance with prediction generated using majority vote inference.

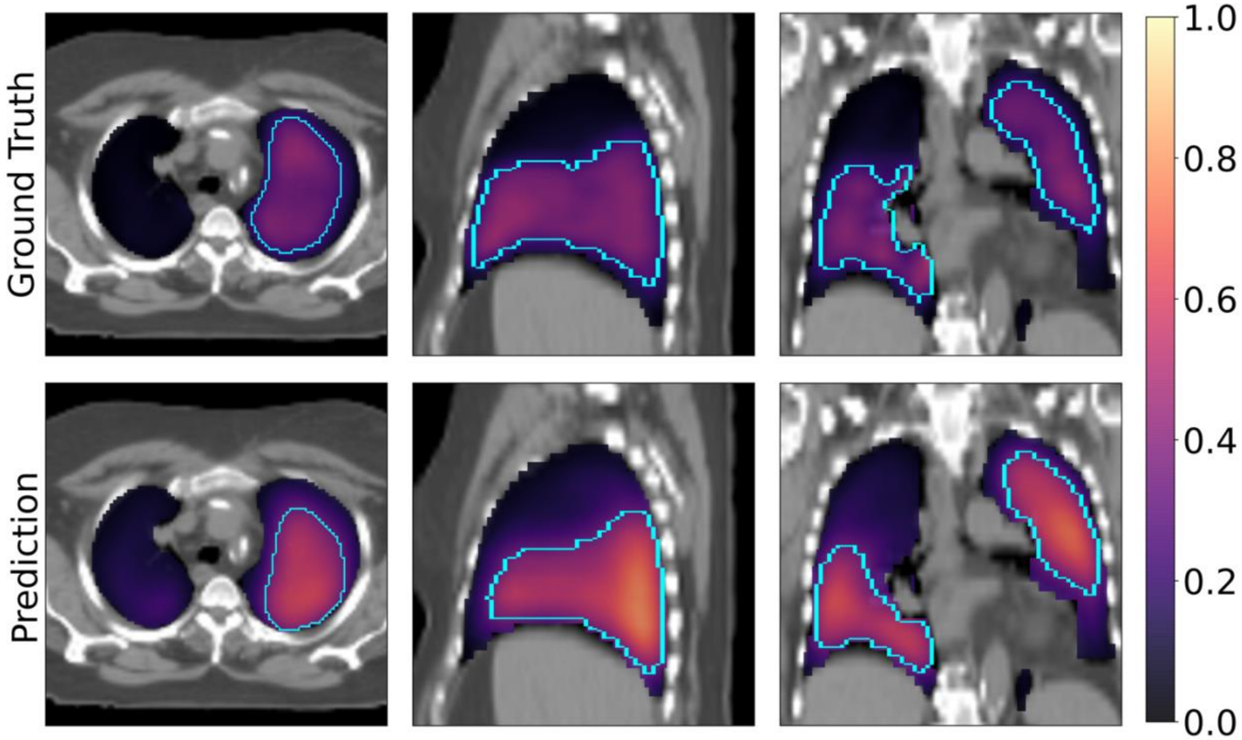


Figure 6.8: The test set study (30_1) with the highest performance with prediction generated using majority vote inference.

6.4 Discussion

Mirroring algorithmic development trends in computer science, the use of statistical learning approaches to create synthetic functional imaging has seen renewed interest with the advent of deep learning architectures. The preponderance of existing synthetic functional imaging literature present mathematical model-based heuristics to create synthetic images. In contrast, a deep learning approach only presupposes that a solution to the task exists and that a signal is present within the data set [49]. To solve a given task, a blank statistical framework, called a model, is trained until it converges upon a robust solution. Thereby, a deep learning technique surpasses many of the shortcomings of traditional problem solving, namely assumptions limited by human perception and the complexity required to devise a robust analytical solution [49]. However, it also presents unique drawbacks in terms of reproducibility and interpretability.

To contrast the deep learning and heuristic approaches for synthetic perfusion creation, we can compare our results to the heuristic method used to calculate perfusion images from 4DCT presented by Castillo *et al.* (2021). In their investigation, Castillo *et al.* generated synthetic perfusion images using a deformable image registration and integrated Jacobian technique to identify local changes in blood mass represented by changes in Hounsfield units between the inhale and exhale image phases. In their study, Castillo *et al.* evaluated their mathematical model on a subset of the validation studies used in this investigation. They reported a Spearman correlation with a median of 0.57 (IQR = 0.305). Using a single-sided Mann Whitney U test, we can compare the results of Castillo *et al.* to the Spearman correlation for our cross-fold validation ($p < 0.001$) and majority vote test set ($p = 0.0749$).

Despite training upon less pre-processed data, our method generates images that can be utilized to generate functional avoidance volumes with 4 of 5 test set patients having Dice similarity coefficients exceeding 0.7, indicating strong correlation [157]. Likewise, the F-score for 3 of 5 of our generated test-set predictions, with a median of 0.803 (IQR: 0.750 – 0.810), perform equivalently to the median F-score of 0.8 for a human observer in repeated segmentation [158]. From our predicted perfusion volumes, we demonstrate the potential application of this technique in the delineation of well-perfused lung for functional avoidance treatment planning. Although further retrospective dosimetric analysis and prospective clinical investigation is required to determine the clinical utility.

Within our test set, patient 32 represents a significant deviation in correlation from the other test set patients. At the time of their pre-treatment imaging sessions, patient 32 presented with emphysema and polycythemia, both of which are known to induce changes in SPECT perfusion [159,160] which potentially contributed to the hypo-perfused posterior region of the right lung (seen in Figure 6.7). The diminished performance observed for this patient indicates a current limitation for the clinical applicability in patients with lung cancer and chronic obstructive pulmonary disease. This is not unexpected though for a deep learning task with a limited training dataset which likely fails to represent all co-morbidities. Therefore, we expect this limitation can be addressed by expanding the training dataset to represent various emphysematous changes. Additionally, if this problem is revisited with an expanded dataset, we intend to report the co-morbidities represented in the training set to transparently declare the expected in- or out-of-domain patients for which our methodology would be applicable.

Radiation-induced injury to lung tissue correlates to a reduction of regional lung perfusion [161]. In a subsequent study, we intend to investigate whether our predicted post-RT perfusion is consistent with the observed radiation-induced reduction in perfusion. An extension would be the investigation of prognostic tools, such as a model capable of predicting post-RT perfusion when given a pre-RT 4DCT and dose distribution. Additionally, performance gains may be achievable through model distillation [162,163] to segment lobar fissures in the lung parenchyma to enforce anatomic boundaries on predicted defects.

Although most functional lung imaging studies use SPECT imaging, it is not without setbacks: availability, cost, and image quality. While our approach overcomes the availability and cost limitations, the predicted image quality is limited by the resolution of MAA-SPECT. Harnessing the improved spatial resolution of ^{68}Ga-MAA or ^{68}Ga-aerosol (Galligas) PET perfusion imaging could increase the resolution of our predicted synthetic images. Similarly, our deep learning technique could be trained on a high-quality ventilation imaging, such as Technegas (Cyclomedica, Kingsgrove, AU). In doing so, a well-trained deep learning model could disseminate the benefits of these limited and costly modalities for a fraction of the cost.

6.5 Conclusion

Our work demonstrates an end-to-end deep learning model that predicts perfusion as demonstrated by statistical correlation and a pragmatic demonstration of generating well-perfused lung contours, which may enable the widespread adoption of perfusion-based functional avoidance radiotherapy planning. The work presented in this chapter was published in a peer-reviewed journal article in 2021 [164].

CHAPTER 7 Open-Source Toolkit and Dataset

7.1 DICOManager: An Open-source Data Processing Toolkit

Digital Imaging and Communications in Medicine (DICOM) was created by the National Electronics Manufacturers Association (NEMA) in 1983 to unify the existing, and diverse, file formats used by the medical device manufacturers. To create a single standard, NEMA needed to allow manufacturers flexibility within the DICOM constraints to support legacy systems without acquiring re-approval from the FDA. Therefore, the DICOM standard can be wide-sweeping, non-uniformly implemented and have multiple pathways to achieve any of the core tasks. Furthermore, DICOM does not necessitate compliance within the header fields which describe the associated image data. These factors often make the translation of code from single- to multi-institution difficult when the DICOM compliance unexpectedly changes. From the experience garnered during the aforementioned deep learning projects, I designed a toolkit for the organization, sorting and processing of DICOM files.

7.1.1 Anatomy of a DICOM

The DICOM standard supports multiple specialties and file subtypes, but the scope of DICOManager [113] is limited to CT, MR and nuclear medicine images, as well as the Radiotherapy (RT) sub-group of the DICOM standard, which covers treatment plans, structure sets and dose files. In the current implementation, DICOManager does not process data in relation to information stored in an RT plan file, therefore we will disregard it from the discussion.

A DICOM image (CT, MR, NM, or PET) is fundamentally comprised of two basic parts: a header and image data. The DICOM header describes patient information, relevant image acquisition parameters, institution information, coordinate systems and conversion factors for the pixel values. The pixel data is then stored in a 2D- or 3D-array of 16-bit integer values which correspond to a given HU-value, SUV or unitless value, depending on image type. Due to the age of the format, design choices were made to limit the memory and computational footprint of DICOM files. For this reason, it is most common for image volumes to be stored as a collection of 2D axial images, each corresponding to a given location in the so-called patient coordinates. The patient coordinate system is attached to the DICOM file at the time of image acquisition and is defined by the imaging system dimensions and coordinates, which are usually in millimeters.

Segmentations generated during clinical treatment planning are saved in DICOM RT structure set. To reduce the file size of the RT structure set, each segmentation is reduced to individual axial slice alpha shapes, which are the minimum number of vertices required to reconstruct the surface of the axial slice. Each of the alpha shape vertices are then saved relative to the patient coordinate system, pointing to the unique identifiers to each of their corresponding axial images. Assembly of the RT structure set can prove tricky because the individual vertices do not necessarily fall on a given rasterized voxel index in the pixel array. Additionally, the RT structure set header lacks the necessary information to determine the original image volume dimensions (in voxels), which would be required to project the assembled segmentation into the volumetric array. Therefore, the reconstruction of an RT structure set is reliant upon the prior interrogation and reconstruction of the corresponding image volume. To account for these unique DICOM design choices, the accurate 3D reconstruction of a given patient's image, segmentations and dose volume necessitate a careful organization of the files for efficient reconstruction.

7.1.2 Assembly of DICOMs

One point of difficulty when translating a deep learning technique to a multi-institution dataset is when 2D slices are lost in the data transfer. This either results in gaps in the assembled 3D volumes or creates volumes which inaccurately represent patient dimensions. Therefore, DICOManager is designed to reconstruct image volumes relative to the patient coordinate system, ensuring that missing slices do not result in assembled volumes of smaller dimensions than what was originally acquired. DICOManager allows users to interpolate or extrapolate contours and image volumes to account for any missing data. For patients with mixed slice thicknesses, interpolation can also be used to generate a single-slice thickness image throughout the volume.

In addition to axial 2D image volumes, DICOManager supports assembly of RT structure sets. Originally designed when computer storage was at a premium, the DICOM format stores a 3D contour as a list of 2D alpha shapes, which contain the minimum number of vertices required to encode the surface of a contour. For contours with inner and outer surfaces, the NEMA 8.8.6.3 specification dictates that a narrow keyhole should be used to join inner and outer surfaces into a single alpha shape. For use in deep learning, structure sets are most useful when assembled as 3D Boolean arrays with equivalent frame of reference as their corresponding image data.

Most contemporary nuclear medicine and PET imaging data is encoded as a single 3D pixel array, making reconstruction simple. The only additional steps required to generate a useful volume is to scale the raw pixel array by the requisite header fields to achieve a quantitatively useful image (e.g., SUV). Computing SUV may require adjusting for the isotope decay time from administration to imaging and the body weight of the patient.

Dose volumes are the last major format supported by DICOManager. Due to the computational demands of computing a dose array, the coordinate systems used are usually coarser than image volumes or have smaller dimensions than the image volume. To achieve voxel-to-voxel correspondence between the dose and image coordinates, a bi-linear interpolation can be used to interpolate the dose to the image grid. For regions of the image volume which are outside the computed dose grid, zeros can be used as filler. Then, from the interpolated dose grid we can use the grid scaling factor header to convert the dose to units of Gray.

7.1.3 Disassembly of DICOMs

For deep learning segmentation tasks, the conversion of a predicted Boolean mask back into RT structure set is critical for a clinical utility. DICOManager supports three methods of converting a Boolean mask back into an RT structure set. The first approach allows a user to append a new structure to an existing structure set without removing the preexisting contours. The second approach allows users to provide a reference structure set, from which a new, uniquely identified structure set can be created with the Boolean mask encoded as a contour. Lastly, a user can generate a new, unique structure set when provided a series of axial CT images.

For each method, the Boolean mask encoding begins by determining each referenced CT image slice and storing the UID references in the DICOM appropriate format. Then each axial slice of the 3D contour is converted to an alpha shape and unraveled into the list of vertices. Finally, the DICOM header is created or updated as needed before the new RT structure set is saved to disk and sorted into the existing cohort group.

7.1.4 Further Work

In its current state, DICOManager has become cumbersome to organize and build upon due the complexity of the tree structure. The only sensible progression of the project is to refactor the code to use an imbedded database for file organization and querying. The default python interpreter comes bundled with an SQLite3 imbedded database, making it the most logical choice for this application. Furthermore, in its current formulation, DICOManager saves assembled image volumes as pickled Numpy dictionaries. Because pickled python objects are inherently executable when read, they pose a substantial security risk and should be replaced with an alternative during the refactoring. An ideal alternative would be to utilize XArray objects, which are Numpy arrays with an attached coordinate system and metadata. Implementing XArrays in the DICOManager backend would vastly simplify the data structure organization and handling. Following the code base refactoring, user accessibility could be improved by increasing the project documentation and bundling the project as a python integrated package manager (pip) project, allowing for single command installation and native execution.

7.2 Publication of Dataset through The Cancer Imaging Archive

7.2.1 Purpose

The Cancer Imaging Archive (TCIA) is a National Cancer Institute funded program, managed by the Frederick National Laboratory for Cancer Research. The TCIA repository was founded with the goal of hosting anonymized, large scale, publicly accessible medical data to facilitate the open collaboration and progress of medical research. Publication of a dataset begins with the proposal to a TCIA steering committee who determines the scope, value, and content of newly accepted data collections. If accepted, the uploading institution and TCIA agree to a data transfer agreement and the terms of data usage. Following the legal paperwork, the collected data set is organized, anonymized, and uploaded to TCIA servers for curation. Curation consists of first ensuring the integrity, uniformity, and completeness of the DICOM images and header tags. During this process, the DICOM header tag anonymization is checked to ensure no patient identifiable information (PHI) still resides. Images are then manually and individually checked to ensure no burned-in PHI (typically left by the reading radiologist) still exists within the image volumes. Following manual curation of both the DICOM header and image, the entire data set is sent to a second, separate curation team who repeats the process. Then, after the extensive check for anonymization, the dataset is ready for public hosting on the TCIA portal for public access.

Having collected data for the hippocampus segmentation work while waiting to receive the RTOG-0933 data, the decision was made to open-source our internal data set. Open sourcing this data has the benefit of allowing for outside validation of our results, facilitates other researchers to build upon our methodology and provides high-quality images for the use in other deep learning tasks. In addition to the images and hippocampus segmentations, the TCIA steering committee asked specifically for the Gamma Knife treatment planning data and potential collaborators requested we also upload any follow-up imaging studies patients had received.

Gamma knife planning data was collected and exported from the Beaumont Gamma Knife Center's Gamma Plan (Elekta AB, Stockholm, Sweden) treatment planning system. For each patient, the DICOM radiotherapy module files for plan, dose and structure set were exported in duplicate for each of the image series used during planning (CT and MR sequences). The structure names used in the Gamma Knife treatment planning files were not modified from their original state, thereby preserving any clinical importance or relevance of the contour naming scheme used for a particular patient. Aside from ensuring registration to the proper imaging sequence (discussed in Section 7.2.3), no modifications were made to the RT plan or dose files either.

Follow-up imaging studies were collected from Beaumont's Philips (Philips Healthcare, Andover, MA) picture archiving and communication system (PACS). Any follow-up MR imaging studies up-to two years after the patient Gamma Knife treatment, or their next Gamma Knife treatment, were collected for this data set. In total, this collection is comprised of 390 patients who presented with vestibular schwannoma (VS, n=73), trigeminal neuralgia (TGN, n=119) or metastatic disease (M, n=198) and were subsequently treated with Gamma Knife (Eleka AB, Stockholm, Sweden) stereotactic radiosurgery. For each patient, the treatment indication is designated with a suffix on their patient ID (VS, TGN or M).

7.2.2 Dataset Composition

All patients in the data set are provided with at least one high-resolution (1 mm slice thickness) T1 FLASH trans-axial MR imaging study and their corresponding high-resolution axial planning CT. When available, treatment planning data (struct, dose, plan), alternative MR sequences (FLAIR, T2 CISS, etc.) and follow-up MR imaging studies were collected. Each MR image used during treatment planning was registered to the CT frame of reference and is provided with the DICOM registration file and the aligned secondary image. Additionally, for each patient in the cohort, hippocampal contours generated by multiple institutional observers are provided in a separate structure set. The total contents of the dataset published on TCIA are given in Table 7.1. Appendix Table D.1 is also provided in the appendix with the top 100 most common, case-insensitive names of each region of interest in the treatment planning structure sets.

Table 7.1: Composition of dataset by DICOM file type.

DICOM FILE TYPE	COUNT
CT	390
MR	3901
REG	872
DOSE	928
PLAN	928
STRUCT (PLANNING)	931
STRUCT (HIPPOCAMPUS)	390

7.2.3 Dataset Preparation and Organization

Consistent organization and DICOM labelling are vital to ensuring that the data is easily accessible to future researchers. The original contour names generated during treatment planning were grouped into 219 categories to improve data accessibility, but care was taken to best preserve clinically relevant data while limiting the groups to a reasonable number. Renaming coverage of 99% (9044/9130) of structures was achieved with the 219 groupings. While TG-263 convention was used, when possible, most tumors were named by their specific anatomical location, which did not have definitions in the TG-263 standardize nomenclature.

Prior to uploading the data to TCIA, measures were taken to ensure consistent DICOM structuring. While most of the Gamma Knife planning data exported from the Gamma Plan treatment planning system was consistent with the DICOM standard, a subset contained many critical flaws. These flaws included inconsistent DICOM unique identifiers (UIDs) which caused DICOM viewers, like MIM Software, to incorrectly read the 3D volumes as a time-series sequence of 2D volumes. Additional important and relevant image acquisition information had also been removed from the DIOCM image files. Flaws in the series continued to the plan, dose, and structure set files which had been stripped of the x-, y-axis patient coordinate systems. This inconsistency in DICOM headers and coordinate systems resulted in patient information which would be effectively unusable by future researchers. Fortunately, the original, unadulterated imaging studies used for the treatment planning were maintained in the Philips PACS. Unfortunately, the images within Philips PACS had different UIDS than the original and required careful pairing and reassociating the data to maintain integrity. To achieve this, a Python script was created to transfer the UID references of the planning data to the original Philips PACS images.

After fixing the broken planning studies, additional steps were taken to create consistency among the cohort. For the imaging data these steps included: renaming one-off MR and CT series descriptions to a more common equivalent description, image series with mixed UIDs were unified under a consistent UID, references to Beaumont Hospital and location were removed and referring physician and operator initials were stripped from the file header. For the hippocampus research contours, the unused hippocampus contours were removed, the study description was changed to "Hippocampus Research Contours" to indicate the structure set was used for hippocampus research and the initials of the contouring individual were replaced with an anonymized alternative. On each of the planning file subtypes, a study description of “Gamma Knife Planning Data” was added to easily indicate the file was used for gamma knife treatment planning, the diagnosis tag was checked for consistency between all plan files, and operator initials were removed from the header. All follow-up imaging studies were processed equivalently to the planning CT and MR imaging studies. Additional processing was the performed, including setting the study description tag to “Follow-up Imaging Set #” to sequentially denote the patient’s imaging studies, any secondary and projection images were removed from their respective series and references to specific hospitals and departments were stripped from the header tags.

After ensuring the integrity of each DICOM file, additional steps were taken to improve the usability of the data. For each patient, MIM (MIM Software, Beachwood, OH) was used to generate a rigid registration between CT and each MR sequence and the registration accuracy was validated using the stereotactic frame fiducial markers. From each registration, a DICOM RT REG file and aligned secondary image volume are provided, with each aligned secondaries series indicated by “[original series description] Co-registered to CT” in the series description DICOM header tag. During export from GammaPlan (Eleka AB, Stockholm, Sweden), the treatment planning files was provided in duplicate for each imaging modality frame of reference (CT and each MR sequence).

In total, 197 patients are provided with follow-up imaging studies, with a median of 2 (range 1-13) follow-up studies provided per patient. A distribution of the number of patients and series in the follow-up imaging studies is provided in Figure 7.1. Each follow-up date exists in a unique frame of reference and was not co-registered to the original treatment planning CT volume.

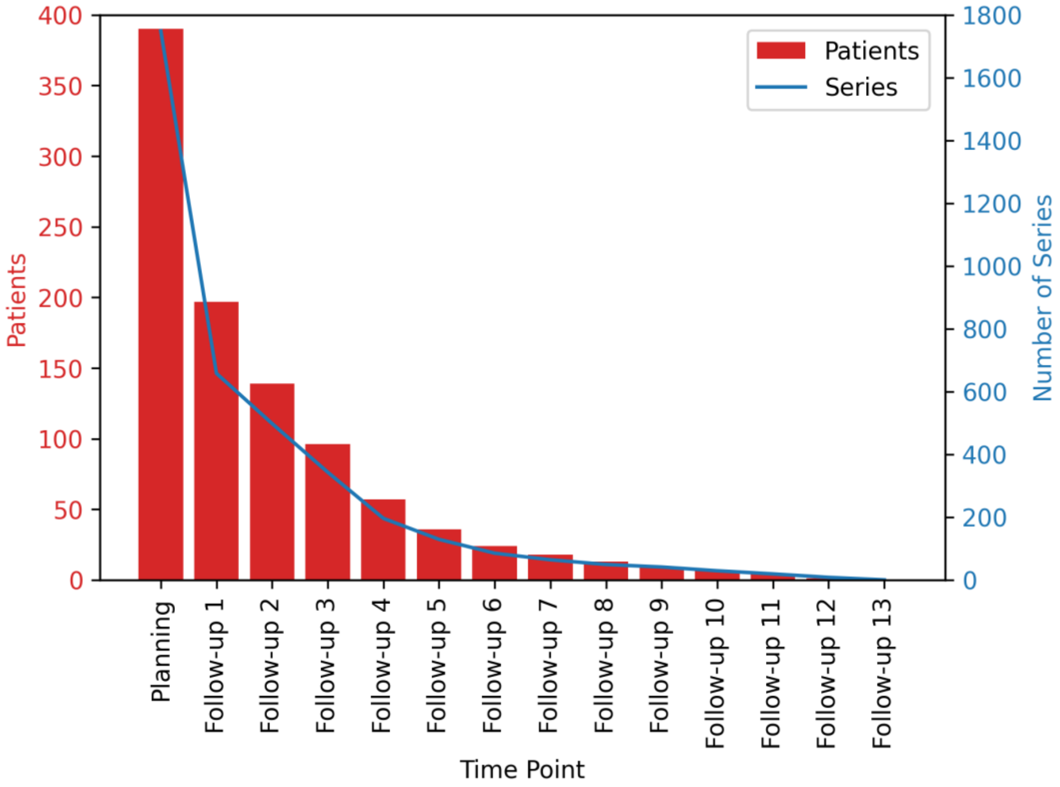


Figure 7.1: Number of patients (red) and series (blue) that are from follow-up imaging studies.

Three independent observers generated hippocampal contours from the CT aligned-secondary of the T1-weighted MR image, with the resultant contours saved to the CT frame of reference. In total, 744 unique left, right contour pairs were generated (observer 1, n=390; observer 2, n=247; observer 3, n=107). In addition to hippocampal contours, the region grow tool was used to generate a head contour (ROI name ‘head’) to mask out the stereotactic frame and remove most of the reconstruction artifacts on the inferior extent of the image volume.

The dataset will be made available through TCIA’s data download portal following curation (expected Q3 2022). Availability of the dataset is expected to coincide with a published manuscript describing the dataset.

CHAPTER 8 Future Work

8.1 Deep Learning Perfusion Validation

In the techniques presented in Chapter 5, it is possible that the deep learning model was not extracting the pulmonary perfusion signal and was instead guessing and getting lucky on the perfusion images. Unfortunately, with only a five patient test set, this scenario could not be ruled out. Therefore, before further work can be done, additional model validation should be conducted. One way would be to see if the model could generate perfusion images which agrees with Galligas PET perfusion imaging, a superior modality for accuracy and spatial resolution. Higher spatial resolution imaging would provide higher certainty that the defects are being accurately identified. Additionally, if the neural network is accurately extracting the pulmonary perfusion signal from the 4DCT, we would expect the generated synthetic images to be consistent with the observed radiation induced damage [165,166]. Additionally, we would expect that a model trained with the proper loss function selected (sensitive to small perfusion defects) would be able to detect the perfusion defects caused by pulmonary emboli. Although, the detection of pulmonary perfusion emboli may require further loss function adjustments as the defects tend to be much smaller.

If any of these investigations proved the signal extracting ability of the neural network, the next step would be to investigate the planning utility of these synthetic pulmonary perfusion images. The images used in this investigation were also used during a functional avoidance lung trial which showed positive results [167], which means the functional avoidance regions and resulting treatment plans are proven to minimize radiation induced damage to lung tissue. The proven clinical outcomes of these contours make for an ideal comparison in treatment planning. If the resultant treatment plans generated comparable functional lung avoidance regions and treatment plans, it would indicate the potential clinical utility of this technology for functional avoidance. If that were the case, it would indicate that further validation through prospective clinical trials may be warranted.

8.2 Hippocampal Avoidance Clinical Application

From the start, the goal was to clinically implement deep learning segmentation for hippocampal avoidance. To achieve this, a robust and stable version of both the trained deep learning model and DICOManager is required, thereby allowing the model to function hands-off at any institution. Once we obtain a stable model validated dosimetrically against RTOG 0933, the intent is to validate the implementation of the technology at an external institution. The most reasonable and safe means of implementing would be to allow physicians to generate contours during regular treatment planning. Each treating physician could then compare and ensure they feel confident with the technology during their regular clinical workflow. This way by the time a patient with a contra-indication for MR imaging, or whose clinical needs necessitate a rapid start to their treatment, the physician will have confidence in the deep learning model’s contour quality. At this point, a manuscript could be prepared from the perspective of a treating physician in the ease of implementation, quality of treatment planning and changes in patient logistics.

Given enough time being implemented in a clinic with traditionally planned patients, and potentially any re-training upon detection of any gross failures, sufficient confidence will be built in the technology. From this point, explorations into the implementation of the methodology into clinics with less accessibility to MR imaging can be made.

8.3 Clinical Trial Quality Assurance with Deep Learning

In addition to clinically implementing the deep learning methodology for clinical care, additional explorations into its utility for quality assurance will be made in future work. The difficulty of hippocampal segmentation is well documented in the clinical trials and the ongoing adoption of HA-WBRT will predominantly be by physicians who did not participate in the trial. Therefore, many of these physicians will not have participated in the contouring workshops, pre-enrollment validation, and treatment plan feedback that the trial participating physicians received. For physicians and patients who do have easy access to MR imaging, there is still the concern that their treatment plans will be non-ideal due to sub-standard hippocampal contours. For this reason, a clinical QA tool which can detect systemic contouring bias would allow for clinics new to treating with HA-WBRT to validate their contouring style. This QA tool could run in the background and compare treating physician contours to the deep learning model prediction. When a certain level of disagreement is achieved with the model, the physician could be notified there may exist and contour style discrepancy and direct them towards reference material, if desired.

The difficulty of this project comes from quantifying contour disagreement in a robust manner. One proposed means of achieving this is to leverage the deep learning model feature space. In attempts to understand what image features deep learning models use to generate segmentations, a new class of research in “explainers” has grown in popularity. In essence, an explainer model identifies what regions of the model are activated by which portions of the image to generate a segmentation of a particular class. If we could take this concept and invert the deep learning model, we could see which activations would generate a particular segmentation. With the activations for any given segmentation, we could compare the activation space between two segmentations to quantify their agreement free from the existing analytical metrics, like Dice and Hausdorff. Through comparison to our planned cases, a threshold of disagreement in contour space which strongly correlates to dose disagreement could be identified. This would mean that when applied to contour quality assurance, we could detect systemic bias not from Hausdorff distance or Dice coefficient (which may or may not correlate to dose), but a deep learning derived, contour-site specific metric which strongly correlates to dose. That way when systemic bias is identified and the contouring style is adjusted, the treatment plans will be brought in line with NRG-CC001 and patients will receive the same improvement in their quality of care.

APPENDIX A

A.1 Common Loss Functions

A.1.1 Mean Squared Error

Mean squared error (MSE) is one of the simplest loss functions used in deep learning. MSE is computed by the mean of the squared loss between the prediction and ground truth, given by Equation A.1.1.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(X_i - Y_i)^2 \tag{A.1.1}$$

Throughout the chapter, we will denote X as the prediction and Y the ground truth. For each tensor, X and Y, there exist *n* classes, encoded as channels. Because it does not require multi-class or multi-label segmentation input, MSE is applicable to both prediction types. However, unlike other functions (e.g., Dice loss), the MSE loss scores during training are not correlative to common segmentation comparison metrics, limiting MSE's overall interpretability. While MSE is an acceptable loss function for certain situations, many more specialized loss functions are available for segmentation tasks. Mean squared error has a built-in implementation in TensorFlow and PyTorch.

A.1.2 Cross Entropy

The term "cross entropy" describes a family of logarithmic loss functions, typically referring to one of two types: binary cross entropy and categorical cross entropy. For both functions, they follow the same basic formula, as given by Equation A.1.2, but differ by the expected input prediction type.

$$CE = -\sum_{i=1}^{n} Y_i \log(X_i) \tag{A.1.2}$$

A.1.3 Binary Cross Entropy

Binary cross entropy is a logarithmic loss function designed for multi-label problems, where the data is limited to a binary value designating class membership. This is most utilized with ground truth data which has been one-hot encoded. This is paired with a model with a sigmoid function as the final activation, providing an output vector with values from zero to one. To reiterate, a multi-label problem would be a task which has volumetrically overlapping segmentations. An example of this is the BraTS Challenge MRI dataset, a brain lesions dataset with structure for the enhancing tumor (ET), tumor core (TC) and whole tumor (WT). To allow for predictions with overlapping structures, the model should output a three-channel mask, with each channel corresponding to one of ET, TC or WT. Binary cross entropy has native implementations in TensorFlow and in PyTorch.

A.1.4 Categorical Cross Entropy

Like binary cross entropy, categorical cross entropy is a logarithmic function. Categorial cross entropy is designed to work with multi-class problems and is compatible with models that have softmax final activation. These models then predict the certainty that any given voxel belongs to each class. Typically, multi-class problems are most useful for segmentation tasks which do not have overlapping classes, such as segmentation of either left or right lung. To prepare the ground truth and model for a softmax activation, the output should have $n + 1$ channels, where n corresponds to the number of segmented structures. This leaves an additional channel to correspond to voxels which are not a member of any class, henceforth referred to as the background. Categorical cross entropy has native implementations in TensorfFlow and PyTorch.

A.1.5 Dice Loss

The Sørensen-Dice coefficient, commonly referred to as the Dice coefficient, was developed for biostatisticians to determine the similarity between two populations [96]. Although it was originally designed to work with tabular binary data, it has proven to be a useful tool for binary segmentation analysis as well [97]. For a set of two contours, the prediction (X) and the ground truth (Y), we can determine the Dice coefficient with Equation A.1.3.

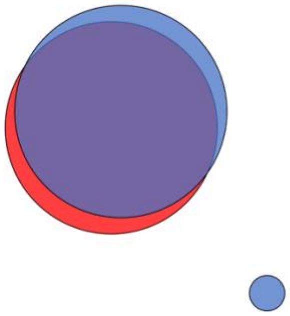
$$Dice = \frac{2|X \cap Y| + \ \varepsilon}{|X| + |Y| + \ \varepsilon} \tag{A.1.3}$$

Where ϵ represents a small value to prevent from having zero division errors when both X and Y are empty and to ensure that Dice = 1 in that instance. Then, the Dice loss function is simply 1 – Dice coefficient, or the negative of the Dice coefficient.

An implementation of the Dice coefficient and Dice loss in Python code using Numpy, Keras and PyTorch are included. You may notice that each implementation looks somewhat different. This is partially because of the different functions and syntax of each library, but also because both the PyTorch and Keras implementations are designed to work with non-binary output data during the training process.

A.1.6 Hausdorff Distance Loss

With the previously discussed loss functions, the prediction agreement was determined by the relative similarity of the structures. This means a well performing prediction could be quite volumetrically accurate but have little penalization for discontinuities in the volume.



Appendix Figure A.1: A visual comparison of two volumetrically similar contours with poor Hausdorff Distance agreement due to the small blue dot having poor spatial agreement with the red circle. In this toy example, red represents a ground truth contour whereas blue indicates the prediction.

Take Appendix Figure A.1 for example, where the prediction is generally in strong agreement with the ground truth, but the prediction also includes a small region with large spatial separation from the ground truth. To translate this to radiation oncology segmentation for treatment planning, while Appendix Figure A.1 has a high Dice coefficient, if this were a target volume, the resulting treatment plan would differ vastly from the ground truth's plan. For radiation oncology treatment planning, an accurate segmentation is primarily one with relatively minimal spatial difference from the ground truth.

To robustly determine the maximum spatial separation between two structures, we can compute Hausdorff distance (HD), as provided by Equation A.1.4.

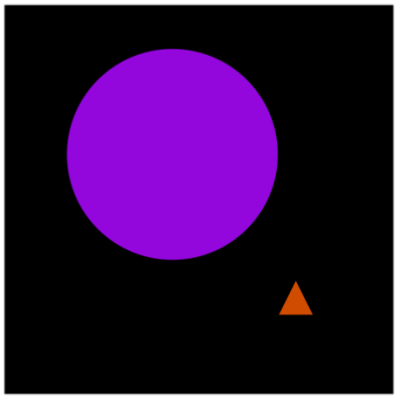
$$HD = \max\{\sup_{y\in Y}\inf_{x\in X} d(y,x), \sup_{x\in X}\inf_{y\in Y} d(y,x)\} \tag{A.1.4}$$

Where sup, inf represent the supremum and infimum of the distances and the distances, represented as d(y,x), are computed between a point from each contour set. Effectively, this metric computes the minimum distance between every point from one surface to two and from surface two to one. Then, the HD is the maximum distance separation from the mappings in either direction, which is the greatest spatial discrepancy between the two surfaces. Unlike previous loss functions, like Dice loss, HD is only dependent on the contour surfaces, meaning a ring and a filled contour could yield the same HD. Thus, HD is particularly sensitive to disjoint segmentations and, when used as a loss function, will reinforce accurate contour boundary predictions. The simplest way to create a HD loss is to compute the negative HD [168]. An implementation of HD is contained within the excellent MedPy library.

Shortcomings of the Hausdorff distance loss function is that it is spatially dependent and highly sensitive to outliers. If the individual image and segmentation masks used to train the model vary in field of view, or have inconsistent voxel dimensions, the HD will be non-uniform across the training set. Most commonly, image voxel dimensions vary in the z-axis, which, if uncorrected for, could yield inconsistent results on the superior or inferior boundaries of a contour. Correction can be achieved by either resampling the image to uniform voxel dimensions or generating the training data sets with corresponding voxel dimensions and passing them into the loss function. Further, the Hausdorff distance metric is sensitive to outliers, which may be overcome by substituting a percentile or mean Hausdorff distance, instead of the traditional total maximum distance.

A.2 Dealing with Class Imbalance

Despite the amazing capabilities of deep learning models, they can also be lazy and will frequently take any available shortcuts to get nearest to the correct answer. So, let us consider the lazy approach to the task of segmenting Appendix Figure A.2 into one of three classes: square, circle and triangle.



Appendix Figure A.2: A representation of an unbalanced segmentation task for two classes (purple, red) with background shown as black.

Between the structures in Appendix Figure A.2, the red triangle is 0.5% of the total area, the purple circle is 23.3% of the total area and the black square is 76.2%. Now, if we are grading the deep learning model's performance with an unbalanced loss function, the deep learning model could omit learning of the triangle completely and be within 99.5% accuracy of the ideal prediction. In this case, if we imagined each shape to represent an anatomical structure, failing to segment any necessary structure would constitute a clinically unacceptable prediction, independent of the accuracy of the other structures.

When choosing a loss function, the task's inherent class balance should be considered to prevent overfitting to only the most dominant classes. While there is no rule-of-thumb for when to choose a balanced or unbalanced loss function, when a task is in fact balanced, the majority of imbalance adjusted loss functions asymptotically approach their unbalanced counterparts. If the significance of class imbalance is unknown, it is recommended to begin with one of the following loss functions.

A.2.1 Weighted Cross Entropy

For tasks with a known and constant magnitude of class imbalance throughout the samples, the imbalance can be compensated for using weighted cross entropy [57]. In similar fashion to the standard cross entropy, weighted cross entropy is a logarithmic function compatible with either multi-label or multi-class data. However, the implementation differs by having a per-class scaling or weighting factor. A cross entropy a binary problem is given in Equation A.1.5.

$$CE = -\sum_{i=1}^{n} w_i Y_i \log(X_i) \tag{A.1.5}$$

Where $\mathrm{w_i}$ represents the per-class weighting to compensate for class imbalance. If $w_i > 1$, then the model will decrease false negatives and if $w_i < 1$ then the model will decrease false positives. To understand false negatives and false positives in the context of segmentation, reference Section 5.e on sensitivity-specificity loss. Weighed cross entropy is natively implemented in TensorFlow and in PyTorch, with the default implementations accepting weights as parameters. An example of this loss function is implemented in A.5.2.

Using a similar implementation as weighted cross entropy, other weighted loss function exist (e.g. weighted Hausdorff distance [169]). Furthermore, it is feasible that any multi-class loss function could be manually adapted to account for class imbalance by including defined class specific weightings.

A.2.2 Generalized Dice Loss

Dice loss is one of the most common loss functions, but it unfortunately is not entirely robust to class imbalances. To account for imbalances, the generalized Dice loss weights the per class Dice score with the inverse square of that class's ground truth volume [98]. This metric is then given by Equation 2.6.

$$GDL = 1 - \frac{2 * \sum_l w_l |X \cap Y| + \varepsilon}{\sum_l w_l |X + Y| + \varepsilon} \quad \text{where} \quad w_l = \frac{1}{Y_l^2} \tag{A.1.6}$$

Where the summation in numerator and denominator represents calculation on a per-class basis. Through inclusion of the weighting factor, the generalized Dice score biases towards classes with a smaller volume proportional to their under-representation.

The strength of the generalized Dice loss function is that it does not require user hyperparameter tuning to compensate for class imbalance. Due to this lack of required tuning, generalized Dice loss is a good imbalance compensating loss function to test first. After experimenting with generalized Dice loss, the other loss functions with greater tunability can be explored to determine if any potential performance gains exist. A NumPy, Keras and Tensorflow compatible implementation of generalized Dice loss are provided in Appendix A.5.3.

A.2.3 No-background Dice Loss

The no-background Dice loss function is a boundary condition of the generalized Dice loss function for when only small structures and the background exist. To address the foreground-background class imbalance, the Dice loss can be calculated on only the structures of interest, excluding the background altogether. During training, the reported loss function is then simply one minus the average Dice coefficient of the structures. Considering that this utilizes the standard Dice coefficient, the loss function only accounts for large imbalances between structures and the background class, not imbalances which may exist between classes. As such, this loss function is best suited for a model concluding with a softmax activation used to segment a single or paired small volume structure.

A.2.4 Focal Loss

Focal loss, as the name implies, adds a focusing mechanism into cross entropy loss which reduces the relative importance of high-confidence predictions [170]. This is particularly relevant for multi-class problems with a final softmax activation where the predictions are certainties of class membership. As the model trains and confidence increases for the membership of certain voxels, those highly confident predictions are down weighted in the loss function. When applied to imbalanced problems, the model will quickly and confidently learn that much of the image volume is a part of the background class. Once this occurs, the focal loss function will shift significance away from the background and on to the accuracy of the remaining structures.

Focal loss achieves dynamic rebalancing by including a scaling factor which decays to zero as probability approaches one. In a simple n-class case, the focal loss is given by Equation A.1.7.

$$FL = -\sum_{i=1}^{n} \alpha_i (1 - X_i)^{\gamma_i} \log{(X_i)} \tag{A.1.7}$$

Where in the equation, X_i is the predicted class membership certainty, α_i is a user adjustable per-class weighting factor and γ_i is a user adjustable per-class focusing parameter. Although, most implementations leave the focusing parameter equal across all classes. When the focusing parameter is $\gamma = 0$, the loss function is equivalent to the cross-entropy loss function. But, as γ becomes larger, the magnitude of focusing increases.

A.2.5 Sensitivity Specificity Loss

To understand Sensitivity-Specificity loss, we should first understand how sensitivity, specificity, as well as precision and recall relate to medical image segmentation. During the calculation of each of these metrics, we consider the voxel-wise accuracy of the segmentations for the evaluation of true / false and positive / negative voxel-wise classifications. Given in Appendix Table A.1 is a description for the four classification classes related to segmentation of either a two-class problem with either foreground or background. Then, from the individual voxel-wise classifications, we can build the definitions of sensitivity, specificity, precision, and recall, as are provided in Appendix Table A.2 with descriptions for the interpretation of the metric.

Appendix Table A.1: Classification types for binary segmentations.

CLASSIFICATION TYPE	DESCRIPTION
TRUE POSITIVE (TP)	Indicates a voxel correctly classified as a member of the class
TRUE NEGATIVE (TN)	Indicates a voxel correctly classified as not a member of the class. Depending on encoding, this may represent the background
FALSE POSITIVE (FP)	Indicates a voxel incorrectly classified as a member of the class, when the ground truth designates it as not a member, or as the background
FALSE NEGATIVE (FN)	Indicates a voxel incorrectly classified as not a member of the class, typically representing background over-prediction

Appendix Table A.2: Metrics to evaluate binary segmentations

STATISTIC	EQUATION	DESCRIPTION
SENSITIVITY OR RECALL	${}^{TP}\!/_{(TP+FN)}$	Represents a model's ability to correctly segment the ROI, with score penalization due to structure under-segmentation, or the prediction of false negatives
SPECIFICITY	${}^{TN}\!/_{(TN+FP)}$	Measures the background segmentation accuracy, with penalization due to ROI over-segmentation
PRECISION	${}^{TP}\!/_{(TP+FP)}$	A measure of a model's capabilities to segment the ROI, with scoring penalization resulting from over-segmentation of the structure, or the prediction of false positives

When we consider the Dice Loss, the function could be represented as the product of recall and precision, as shown in Equation A.1.8.

$$Dice = \frac{2|X \cap Y|}{|X| + |Y|} = \frac{TP}{TP + FP} * \frac{TP}{TP + FN} \tag{A.1.8}$$

If we instead want to calculate sensitivity-specificity loss (SSL) [171], we could adjust the balance between the two terms with a factor, in this case r. This would then give the sensitivity-specificity loss function as in Equation A.1.9.

$$SSL = r \frac{TP}{TP + FP} * (1 - r) \frac{TN}{TN + FP} \tag{A.1.9}$$

Which is computed as a combination of the mean squared errors between the prediction (sensitivity) and the background (specificity), which is provided in Equation A.1.10.

$$SSL = r\,\frac{\sum_i (X_i - Y_i)^2 X_i}{\sum_i X_i} + (1-r)\frac{\sum_i (X_i - Y_i)^2 (1 - X_i)}{\sum_i (1 - X_i)} \qquad \text{(A.1.10)}$$

Where we can account for the background to foreground weighting with the r factor, where a higher value of r places a larger emphasis on the sensitivity, or foreground voxels. Like weighted cross entropy, this loss function allows for user adjustable weighting to compensate for class imbalances present in the task.

A.2.6 Tversky Loss

If our segmentation task requires higher sensitivity to either false negatives or false positives, a variable index, such as the Tversky loss can be utilized [172]. The Tversky index is given in Equation A.1.11.

$$Tversky = \frac{|X \ \cap Y|}{|X \ \cap Y| + \ \alpha|{\sim}Y| + \beta|{\sim}X|} = \frac{TP}{TP + \ \alpha * FP + \beta * FN} \qquad \text{(A.1.11)}$$

Where the ~ operator indicates the relative compliment of the boolean array and the values of α, β are hyperparameters corresponding to magnitude of the penalization for FP and FN, respectively. Through adjusting the ratio of $\frac{\alpha}{\beta}$, the performance of the loss function can be modified. In the instance that $\alpha = \beta = 0.5$, the Tversky loss function become equivalent to the Dice loss function.

The Tversky loss function’s strength is, if the user is so inclined, it can be adjusted to exactly counteract the task’s class imbalance or segmentation needs. For example, segmentation tasks which prioritize ROI coverage could have a lower ratio of $\frac{\alpha}{\beta}$, whereas tasks which require minimal over-expansion of segmentations would utilize a higher ratio.

A.3 Compound Loss Functions

Many of the aforementioned loss functions exhibit unique properties which make them well suited for a particular segmentation task. Occasionally, however, problems require properties at the intersection of multiple loss functions. Fortunately, different loss functions can be combined to span a larger set of properties.

A.3.1 Dice + Cross Entropy

The combination of cross entropy and Dice coefficient is a popular pairing for loss functions [173]. Alone, the Dice coefficient is robust to minor class imbalances, but does not allow for weighting of false positives or false negatives. The two terms within a weighted binary cross entropy function, however, can be modified to increase or decrease the penalty for false negative or false positive values. When dice and cross entropy losses are combined, the result is a partially class imbalanced loss function with variable sensitivity for false predictions.

A.3.2 Dice + Focal Loss

A further example of combined loss function is Dice loss and focal loss [174]. More precisely, their loss function implementation utilized the Tversky loss function with $\alpha = \beta = 0.5$, although these hyperparameters could have been tuned differently for this task. Through the combination, this joint loss function combines both the volumetric dependency of the Dice coefficient and the focal loss property of increased importance of highly uncertain predictions.

A.3.3 Non-linear Combinations

To generate the most utility from a combined loss function, the balance between the terms should exhibit non-linear behavior. A strong loss function combination should choose loss functions which each possess unique properties. For some tasks, these behaviors can be more powerful at the early or late stages of training.

For example, take the Hausdorff loss function. Traditionally, the 100th percentile Hausdorff distance is highly sensitive to spatial outliers which limits the usefulness during early training.

However, this becomes an asset during late training stages, as it can accurately discriminate against spatial outliers, thus fine-tuning performance.

Another example of a potential non-linear combination is Dice and focal loss. In the original loss function implementation, the Dice loss term dominated for epochs with poor validation set performance. Then, the importance of the focal loss term increased as the validation set performance improved. This gradual shift in balance allowed the model to partially train on Dice loss before becoming dominated by focal loss and being penalized for high prediction uncertainty.

It should be noted that non-linear loss function combinations will require additional hyperparameter tuning and are more likely to train inconsistently. A suggested workflow is to begin training the model with only the initially dominant term. Then, once hyperparameter-tuned, the loss function can be expanded with the minor terms, before re-tuning the hyperparameters.

A.4 Dealing with Imperfect Data

For most medical image segmentation tasks, the training data set must be large, diverse, and high quality. Unfortunately, particularly in medicine, creating such a training set is a time-consuming undertaking. This is particularly problematic when the generation of ground truth labels requires an expert, whose time is likely at a premium.

An ongoing field of research attempts to create methods and loss functions to train high quality models from imperfect data. In many clinical cases, only the relevant selection of all organs-at-risk are segmented. This means that the original clinical data set may not be densely populated with all structures on all cases. For cases that lack a labeled structure, gradient backpropagation will penalize a model’s potentially accurate prediction due to imperfections in the ground truth.

A few attempts to account for imperfect data, particularly sparsely labeled ground truths, have achieved success through modification of the loss function. For example, Bokhorst *et al.*[175] trained a U-Net model from sparely labeled histology images by only backpropagating the loss function from channels which had 'valid' ground truth labels. Zhu *et al.*[174] extended this concept by not only masking for only 'valid' ground truths but weighting each class at the inverse of their occurrence. Through doing so, the loss function compensated for the inter-class imbalance deriving from the sparsely labeled ground truth. Although these are promising first steps, the further adaptation of loss functions to train robustly on imperfect data will continue to garner interest for medical image segmentation.

A.5 Loss Function Code Examples

A.5.1 Dice Loss

A python code example of dice loss compatible with the Numpy array library. Code is designed to compute the dice coefficient of two arrays (output, labels) and return the dice loss.

```
import numpy as np

def dice_coef(output, labels):
    # Computes the dice coefficient of two numpy arrays
    eps = np.finfo(float).eps
    intersection = np.sum(output * labels)
    denominator = np.sum(output) + np.sum(labels)
    return (2 * intersection + eps) / (denominator + eps)

def dice_loss(output, labels):
    # Computes the dice loss of two numpy arrays
    return 1 - dice_coef(output, label)
```

A.5.2 Weighted Cross Entropy

Weighted binary cross entropy function, written in Python to be compatible with the TensorFlow Keras library. Weights are specified when the class instance is initialized, and the binary or categorical cross entropies can then be computed with class method calls.

```
import tf.keras.backend as K
```

```
class weighted_cross_entropy:
        def __init__(self, weights):
                self.weights = weights
                self.eps = K.epsilon()

        def binary(y_true, y_pred):
                term0 = y_true * K.log(y_pred) * self.weights[0]
                term1 = (1 – y_true) * K.log(1 – y_pred) * self.weights[1]
                bce = -1 * (term0 + term1)
                loss = K.mean(bce, axis=-1)
                return loss

        def categorical(y_true, y_pred):
                y_pred /= K.sum(y_pred, axis=-1, keepdims=True)
                y_pred = K.clip(y_pred, self.eps, 1 – self.eps)
                loss = -1 * K.sum(self.weights * y_true * K.log(y_pred))
                return loss
```

A.5.3 Generalized Dice Loss

The generalized dice loss function builds upon the previous example of a Numpy compatible dice loss function. Generalized dice loss weights each channel (corresponding to a segmentation class) inversely proportional to the rate of occurrence in the ground truth. Including inverse weighting accounts for class imbalance which would occur when all classes are weighted equally in the standard dice loss function.

```
import numpy as np

def generalized_dice_coef(output, labels):
    # Computes the generalized dice coefficient of two numpy arrays
    eps = np.finfo(float).eps
    sum_dims = tuple(range(labels.ndim))
    w = 1 / (np.sum(labels, axis=sum_dims[:-1])**2 + eps)
    numerator = np.sum(w * np.sum(output * labels, axis=sum_dims))
    denominator = np.sum(w * np.sum(output + labels, axis=sum_dims))
    return (2 * numerator + eps) / (denominator + eps)

def generalized_dice_loss(output, labels):
    # Computes the generalized dice loss of two numpy arrays
    return 1 - generalized_dice_coef(output, labels)
```

A.5.4 No Background Dice Loss

No background dice loss is a simple modifier to the standard dice loss function for highly imbalanced segmentation tasks where the background is orders of magnitude greater than the semantic segmentation classes. Prior to computation of the dice loss, the background channel is excluded from the output (prediction) and label (ground truth) classes. This function is compatible with the Numpy array library.

```
# Requires the standard dice loss implementation as well
import numpy as np

def no_bkgd_dice_loss(output, labels):
    # Computes the dice loss of two numpy arrays
    return 1 - dice_coef(output[…, 1:], label[…, 1:])
```

A.5.5 Tversky Loss

Tversky loss is a derivative of the dice loss function. Dice loss is inherently designed to equally penalized both false positive and false negative segmentation voxels, but this bias is not always ideal for segmentation tasks. Therefore, having the ability to alter the weighting between false positive and false negative predictions may be desired and can be obtained with the Tversky loss function. The example provided is compatible with the Numpy array library.

```
class losses:
    def __init__(self, alpha=0.5, beta=0.5, loss=True):
        self.alpha = alpha
        self.beta = beta

    def tversky(self, output, labels):
        # Calculates the tversky coefficient or loss
        eps = np.finfo(float).eps
        true_pos = np.sum(output * labels)
        false_pos = self.alpha * np.sum(labels * (1 - output))
        false_neg = self.beta * np.sum(output * (1 - labels))

        tversky = (true_pos + eps) / (true_pos + false_pos + false_neg + eps)
        return tversky

    def tversky_loss(self, output, labels):
        return 1 – self.tversky(output, labels)
```

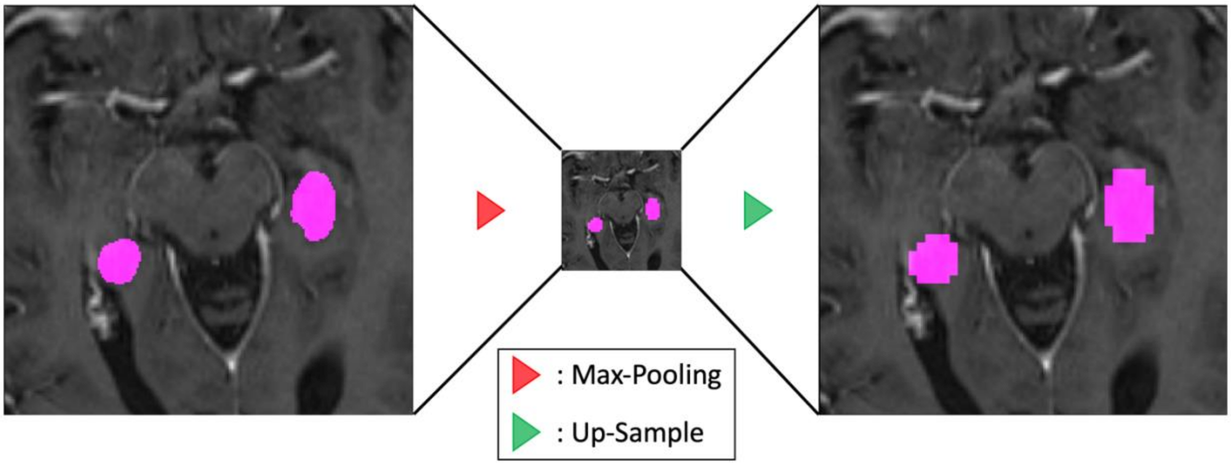
APPENDIX B

B.1 Limitations in Segmentation Accuracy Resulting from Changes in Field of View

In small structure segmentation tasks, cropping the image volumes is vital to the success of a model. Other models designed for segmentation [68,89–91] have utilized image down-sampling prior to prediction, followed by up-sampling to the original image dimensions. To boost their performance, some models follow up-sampling with a conditional random field (CRF) for post processing segmentation improvements [89–91]. Since cranial CT images lack high levels of contextual information and, at small fields of view are noise dominated, utilizing CRFs is ineffective since the process relies on voxel statistics and feature edges to group predicted volumes [176,177]. Further, up-sampling predictions of low volume structures, such as the hippocampus, causes a loss in spatial resolution and significantly limits the achievable accuracy of both the Hausdorff distance and Dice similarity coefficient.

To fully evaluate the impact of changes in field of view on the theoretical limit of segmentations, a toy model was designed to simply down-sample and up-sample the ground truth segmentations (Appendix Figure B.1). These modified segmentations were then compared to the original ground truth to determine the absolute change in Hausdorff Distance (mm) and Dice correlation coefficient (%), as given in Appendix Table B.1. This toy problem showed that a 1:2 change in the field of view would result in an uncertainty of 7.2% Dice score and 0.721 mm Hausdorff score. With the RTOG-0933 criteria set with a hard boundary of 7mm, an uncertainty of 10% was determined to be too substantial in evaluating model accuracy. Therefore, cropping instead of resampling was used during the model perpetration.

While cropping reduces contextual information in the boney anatomy, the RTOG 0933 protocol for contouring the hippocampus defines the inferior, superior, and lateral boarders from the lateral ventricles. Due to the high importance of the lateral ventricles in defining the hippocampus, and their ability to be visualized on a CT image, we chose the crop volume to include the lateral ventricles but reduce cranial anatomy irrelevant to defining the hippocampus. Through cropping, we were able to reduce the model's memory footprint and computational complexity, thereby decreasing the time required to conduct a nested cross-fold validation.



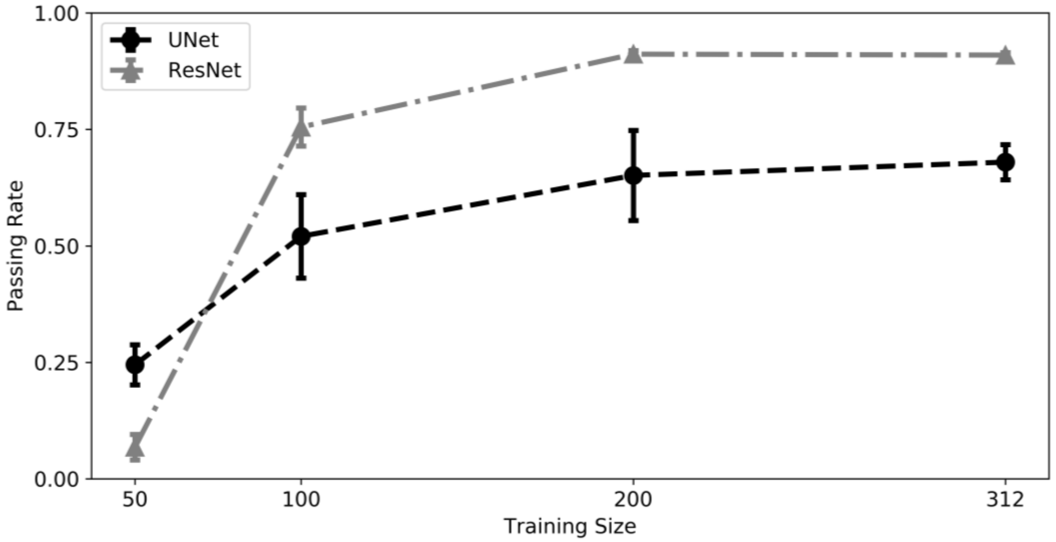
Appendix Figure B.1: Effect of down-sampling on the theoretical performance limit. A contour (magenta) is down-sampled and up-sampled using max-pooling and up-sampling layers implemented in TensorFlow. An MR image is included for visual comparison for a 1:4 sampled contour. The resultant contours differ from the original due to information loss in the max-pooling and pixilation from up-sampling.

Appendix Table B.1: Down-sampled and up-sampled 1:x images with 100th-percentile Hausdorff Distance and Dice limits reported as mean and standard deviation. These values were computed across the entire 390-patient cohort.

DOWN SAMPLE RATE	HAUSDORFF DISTANCE (MM)	DICE LIMIT (%)
1:1	0.000 ± 0.00	100. ± 0.00
1:2	0.721 ± 0.00	92.8 ± 1.30
1:4	2.134 ± 0.08	80.4 ± 2.72
1:8	4.664 ± 0.33	61.6 ± 4.30

B.2 Model Sub-type Saturation

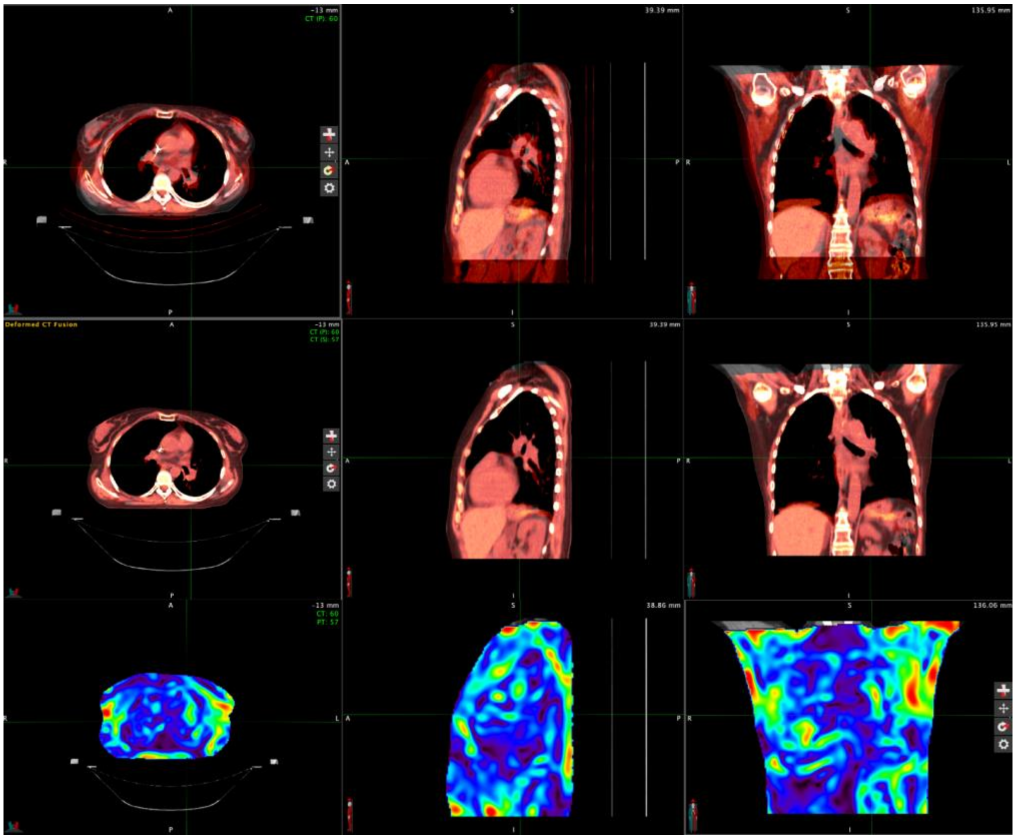
When training a deep learning network, it is difficult to know the requisite dataset size to properly saturate the models used in the experiment. To retrospectively determine if the chosen models saturated adequately, we generated randomly selected subsets of the data (N=50, 100, 200, 312). From these cohorts, we tracked the relative performance on each of the main two model types (ResNet, UNet) to determine if we achieved adequate saturation. These results are provided in Appendix Figure B.2 and show that saturation for both model subtypes was achieved at approximately 200 patients for the ResNet and potentially achieved between 200 and 300 for the 3D UNet. The uncertainty in 3D UNet training performance for the different cohorts is likely attributable to the inefficient method of pooling and transposed convolutions to provide increasing field of view and representation of higher-order features in the 3D UNet. Due to the inefficiencies of this encoder-decoder style model architecture, the 3D UNet is comprised of 30x more parameters than the comparable ResNet architecture. This increased model parameter and layer count, require a larger dataset to fully back-propagate the gradient throughout the model to yield stable and consistent performance.



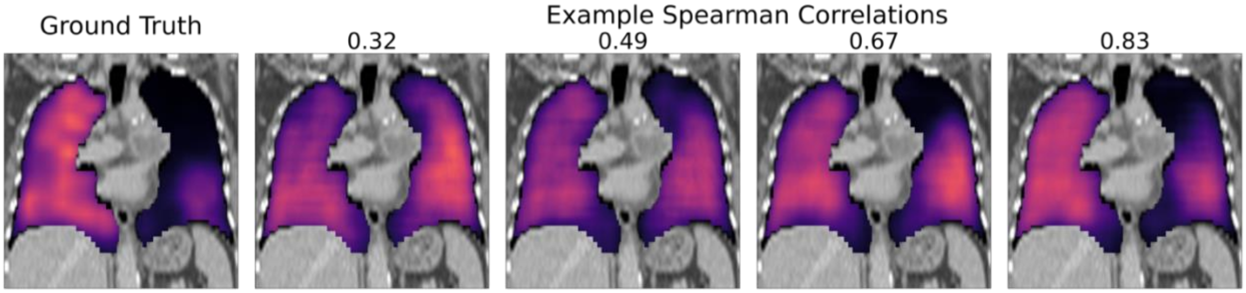
Appendix Figure B.2: Mean passing rate of left, right hippocampus, plotted as a function of training size. Ten models were trained on randomly selected data with the error bars reported at standard deviation of the mean.

APPENDIX C

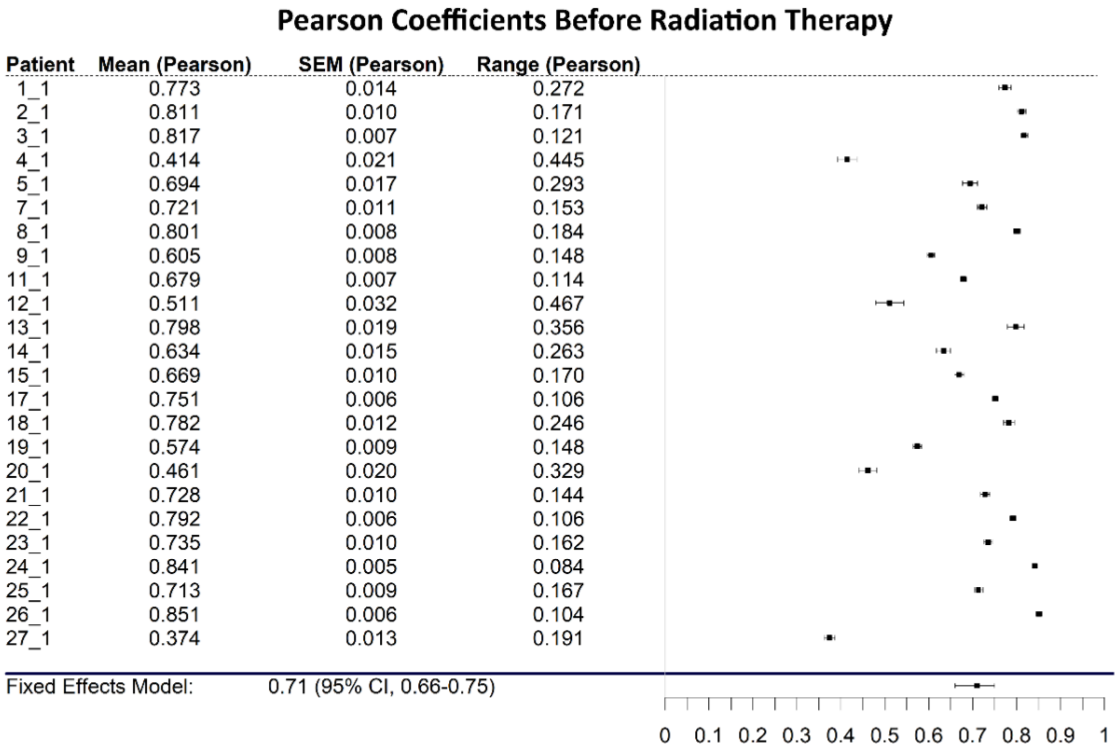
C.1 Synthetic Pulmonary Perfusion Additional Analysis



Appendix Figure C.1: An overlay of the registered SPECT CT and AIP-CT using a rigid registration (top) and deformable registration (middle). The curl of the deformable vector field (bottom) depicts inconsistent and non-physical deformations within the lung parenchyma.



Appendix Figure C.2: Comparison of a distribution of Spearman correlations computed on a single coronal slice relative to the ground truth (study 5_1). The given images are sourced from different trained models during the cross-fold validation.

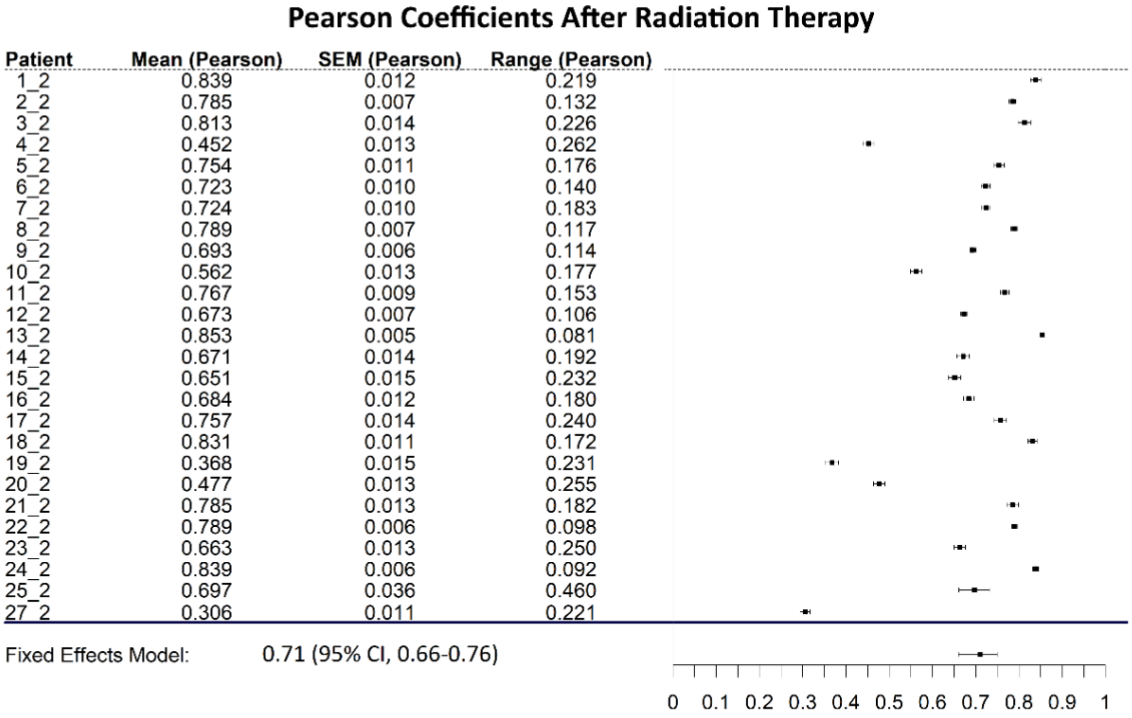


Pearson Coefficients Before Radiation Therapy

Patient	Mean (Pearson)	SEM (Pearson)	Range (Pearson)
1_1	0.773	0.014	0.272
2_1	0.811	0.010	0.171
3_1	0.817	0.007	0.121
4_1	0.414	0.021	0.445
5_1	0.694	0.017	0.293
7_1	0.721	0.011	0.153
8_1	0.801	0.008	0.184
9_1	0.605	0.008	0.148
11_1	0.679	0.007	0.114
12_1	0.511	0.032	0.467
13_1	0.798	0.019	0.356
14_1	0.634	0.015	0.263
15_1	0.669	0.010	0.170
17_1	0.751	0.006	0.106
18_1	0.782	0.012	0.246
19_1	0.574	0.009	0.148
20_1	0.461	0.020	0.329
21_1	0.728	0.010	0.144
22_1	0.792	0.006	0.106
23_1	0.735	0.010	0.162
24_1	0.841	0.005	0.084
25_1	0.713	0.009	0.167
26_1	0.851	0.006	0.104
27_1	0.374	0.013	0.191

Fixed Effects Model: 0.71 (95% CI, 0.66-0.75)

Appendix Figure C.3: Forest plot of the validation set, pre-treatment Spearman correlation coefficients. Encoding of the study is by patient with '_1' representing a pre-treatment imaging study.

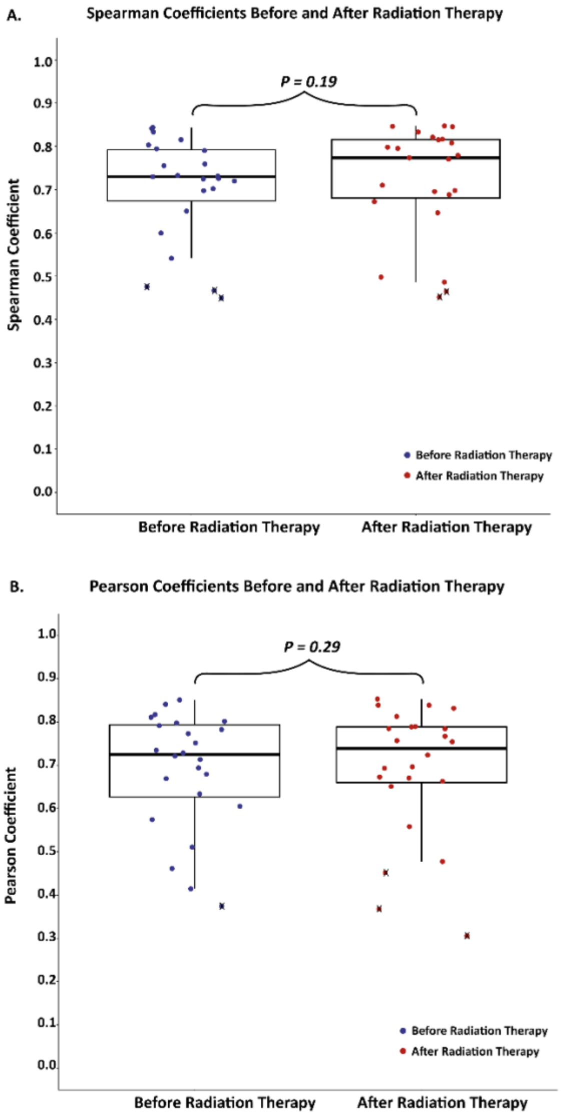


Pearson Coefficients After Radiation Therapy

Patient	Mean (Pearson)	SEM (Pearson)	Range (Pearson)
1_2	0.839	0.012	0.219
2_2	0.785	0.007	0.132
3_2	0.813	0.014	0.226
4_2	0.452	0.013	0.262
5_2	0.754	0.011	0.176
6_2	0.723	0.010	0.140
7_2	0.724	0.010	0.183
8_2	0.789	0.007	0.117
9_2	0.693	0.006	0.114
10_2	0.562	0.013	0.177
11_2	0.767	0.009	0.153
12_2	0.673	0.007	0.106
13_2	0.853	0.005	0.081
14_2	0.671	0.014	0.192
15_2	0.651	0.015	0.232
16_2	0.684	0.012	0.180
17_2	0.757	0.014	0.240
18_2	0.831	0.011	0.172
19_2	0.368	0.015	0.231
20_2	0.477	0.013	0.255
21_2	0.785	0.013	0.182
22_2	0.789	0.006	0.098
23_2	0.663	0.013	0.250
24_2	0.839	0.006	0.092
25_2	0.697	0.036	0.460
27_2	0.306	0.011	0.221

Fixed Effects Model: 0.71 (95% CI, 0.66-0.76)

Appendix Figure C.4: Forest plot of the validation set, post-treatment Spearman correlation coefficients. Encoding of the study is by patient with '_2' representing a post-treatment imaging study.



Appendix Figure C.5: Box and whisker plots of mean Spearman (3.A) and Pearson (3.B) correlation coefficients before and after radiation therapy. Boxes represents the interquartile range (IQR) with the horizontal line representing the median value. The whiskers correspond to 1.5 x IQR above and below the median and the X's represent outliers. Wilcoxon rank-sum test was used to compare median values, P < 0.05 was considered statistically significant.

Appendix Table C.1: Contour comparison metrics for each pre-RT validation set imaging study (N=24), reported as median and IQR.

METRIC	F50 CONTOURS
DICE	0.780 (0.762 – 0.812)
ASD (MM)	6.23 (5.13 – 8.70)
HD95% (MM)	20.9 (15.8 – 26.4)
RAVD (%)	0.28 (-0.21 – 0.75)
PRECISION	0.779 (0.760 – 0.811)
RECALL	0.781 (0.767 – 0.813)
F-SCORE	0.780 (0.762 – 0.812)

APPENDIX D

D.1 TCIA Dataset Planning Contour Names

Given in Appendix Table D.1 is a list of the top 100 most common structure names and their total number of occurrences. Note, the contour listed as “Plan1[tgt#]#gy” is a naming convention used for each target denoted by a letter (a, b, …), and the dose to the target specified as #gy (for Gray). For cases with multiple lesions, relative anatomical directions were common in the contour names. To preserve clinical data, abbreviations were used for: left (L), right (R), medial (med), midline (mid), lateral (lat), anterior (ant), posterior (post), inferior (inf) and superior (sup). Cranial nerves were renamed from their common names (e.g., trigeminal nerve) to their cranial nerve numbering (e.g., CN_V), with treatment contours for the trigeminal nerve are denoted with “TX”. Empty contours are designated in their ROI name with ‘(Empty)’ appended to the end.

Appendix Table D.1: A table of the 100 most common Gamma Knife treatment planning structures.

STRUCTURE	#	STRUCTURE	#	STRUCTURE	#	STRUCTURE	#
Plan1[tgt#]#gy	1421	**Cerebellum_Lat_R**	64	**Lobe_Parietal_Post_R**	33	**Cerebellum**	19
Skull	928	**Cerebellum_Inf_L**	61	**Lobe_Frontal_Ant_L**	33	**Lobe_Occipital_Mid_L**	19
Brainstem	466	**Lobe_Frontal_Ant_R**	57	**Vermis**	31	**Periventricular_Post_R**	19
Brain	453	**Lobe_Frontal_Sup_L**	54	**Periventricular_Post_L**	30	**Lobe_Occipital_Med_R**	18
Lobe_Frontal_R	271	**Lobe_Parietal_Post_L**	50	**Lobe_Temporal_Lat_L**	29	**Tentorium_L**	18
Lobe_Frontal_L	252	**Vermis_L**	46	**Basal_Ganglia_L**	28	**Cerebellum_Sup_R**	18
Cerebellum_L	243	**Lobe_Frontal_Inf_L**	46	**Lobe_Frontal_Lat_R**	28	**Lobe_Occipital_Ant_L**	18
Cerebellum_R	218	**GTV**	44	**Cerebellum_Mid_R**	28	**Pervivent_L**	17
Lobe_Occipital_L	204	**Lobe_Frontal_Inf_R**	42	**Cerebellum_Midline**	28	**Frontoparietal_R**	17
Lobe_Occipital_R	189	**Cerebellum_Lat_L**	42	**Cerebellum_Med_L**	27	**Thalamus_L**	17
Lobe_Temporal_L	175	**Vermis_R**	41	**Lobe_Frontal_Lat_L**	26	**Pons_R**	16
Lobe_Parietal_R	169	**Resection_Cavity**	41	**CN_VII**	25	**Lobe_Temporal_Inf_L**	16
CV_V_TX	163	**Acoustic Neuroma**	40	**Basal_Ganglia_R**	25	**Lobe_Temporal_Lat_R**	16
Lobe_Parietal_L	160	**Cerebellum_Ant_L**	40	**Cerebellum_Mid_L**	25	**CV_V_TX_L**	15
CN_V	157	**Lobe_Frontal_Mid_R**	39	**Lobe_Temporal_Ant_L**	25	**Periventricular_Ant_R**	15
Lobe_Temporal_R	153	**Periventicular_R**	39	**Lobe_Temporal_Med_R**	24	**Lobe_Parietal_Mid_L**	15
Cochlea	134	**Brainstem_Pons**	38	**Lobe_Temporal_Ant_R**	23	**Lobe_Occipital_Inf_L**	15
Labyrinth	119	**Cerebellum_Post_R**	38	**Lobe_Frontal_Med_R**	22	**Insula_L**	14
CN_V_R	98	**Cerebellum_Ant_R**	38	**Lobe_Temporal_Post_R**	22	**Lobe_Parietal_Med_R**	12
Acoustic_Neuroma_L	96	**Lobe_Frontal_Sup_R**	37	**Lobe_Frontal_Med_L**	21	**Lobe_Occipital_Ant_R**	12
CN_V_L	86	**Lobe_Occipital_Post_R**	37	**Lobe_Frontal_Mid_L**	21	**Pons**	12
Lobe_Frontal_Post_R	77	**CV_V_TX_R**	36	**Cerebellum_Sup_L**	20	**Lobe_Frontal_Ant_Lat_L**	12
Lobe_Frontal_Post_L	71	**Lobe_Parietal_Sup_R**	35	**Cerebellum_Med_R**	20	**Vermis_Mid**	12
Acoustic_Neuroma_R	67	**Lobe_Occipital_Post_L**	34	**Frontoparietal_L**	20	**Lobe_Frontal_Ant_Mid_L**	12
Cerebellum_Post_L	67	**Lobe_Parietal_Sup_L**	34	**Thalamus_R**	20	**Tempooccipital_L**	12

REFERENCES

[1] Willis Ware, in *Proc. West. Jt. Comput. Conf.* (The Institute of Radio Engineers, Los Angeles, 1955), p. 85.

[2] D.E. Rumelhart, G.E. Hinton, and R.J. Williams, Nature **323**, 533 (1986).

[3] Y. LeCun, B. Boser, J.S. Denker, D. Henderson, R.E. Howard, W. Hubbard, and L.D. Jackel, Neural Comput. **1**, 541 (1989).

[4] H. Li, Z. Xu, G. Taylor, C. Studer, and T. Goldstein, ArXiv171209913 Cs Stat (2017).

[5] J.-H. Chao, R. Phillips, and J.J. Nickson, Cancer **7**, 682 (1954).

[6] D.F. Young, J.B. Posner, F. Chu, and L. Nisce, Cancer **34**, 1069 (1974).

[7] W.A. Hindo, F.A. Detrana III, M.-S. Lee, and F.R. Hendrickson, Cancer **26**, 138 (1970).

[8] M.L. Evans, M.M. Graham, P.A. Mahler, and J.S. Rasey, Int. J. Radiat. Oncol. **13**, 563 (1987).

[9] B. Borgelt, R. Gelber, S. Kramer, L.W. Brady, C.H. Chang, L.W. Davis, C.A. Perez, and F.R. Hendrickson, Int. J. Radiat. Oncol. **6**, 1 (1980).

[10] H. Aoyama, H. Shirato, M. Tago, K. Nakagawa, T. Toyoda, K. Hatano, M. Kenjyo, N. Oya, S. Hirota, H. Shioura, E. Kunieda, T. Inomata, K. Hayakawa, N. Katoh, and G. Kobashi, JAMA **295**, 2483 (2006).

[11] M. Kocher, R. Soffietti, U. Abacioglu, S. Villà, F. Fauchon, B.G. Baumert, L. Fariselli, T. Tzuk-Shina, R.-D. Kortmann, C. Carrie, M.B. Hassel, M. Kouri, E. Valeinis, D. van den Berge, S. Collette, L. Collette, and R.-P. Mueller, J. Clin. Oncol. **29**, 134 (2011).

[12] A.V. Tallet, D. Azria, F. Barlesi, J.-P. Spano, A.F. Carpentier, A. Gonçalves, and P. Metellus, Radiat. Oncol. **7**, 77 (2012).

[13] W.B. Scoviille and B. Milner, 11 (n.d.).

[14] P.S. Eriksson, E. Perfilieva, T. Björk-Eriksson, A.-M. Alborn, C. Nordborg, D.A. Peterson, and F.H. Gage, Nat. Med. **4**, 1313 (1998).

[15] T.J. Shors, G. Miesegaes, A. Beylin, M. Zhao, T. Rydel, and E. Gould, Nature **410**, 372 (2001).

[16] O.K. Abayomi, Acta Oncol. Stockh. Swed. **35**, 659 (1996).

[17] V. Gondi, S.L. Pugh, W.A. Tome, C. Caine, B. Corn, A. Kanner, H. Rowley, V. Kundapur, A. DeNittis, J.N. Greenspoon, A.A. Konski, G.S. Bauman, S. Shah, W. Shi, M. Wendland, L. Kachnic, and M.P. Mehta, J. Clin. Oncol. Off. J. Am. Soc. Clin. Oncol. **32**, 3810 (2014).

[18] J. Brandt, Clin. Neuropsychol. **5**, 125 (1991).

[19] N. Thavarajah, G. Bedard, L. Zhang, D. Cella, J.L. Beaumont, M. Tsao, E. Barnes, C. Danjoux, A. Sahgal, H. Soliman, and E. Chow, Support. Care Cancer **22**, 1017 (2014).

[20] F.I. Mahoney and D.W. Barthel, Md. State Med. J. **14**, 61 (1965).

[21] M.P. Mehta, P. Rodrigus, C. h. j. Terhaard, A. Rao, J. Suh, W. Roa, L. Souhami, A. Bezjak, M. Leibenhaut, R. Komaki, C. Schultz, R. Timmerman, W. Curran, J. Smith, S.-C. Phan, R.A. Miller, and M.F. Renschler, J. Clin. Oncol. **21**, 2529 (2003).

[22] V. Gondi, R. Tolakanahalli, M.P. Mehta, D. Tewatia, H. Rowley, J.S. Kuo, D. Khuntia, and W.A. Tomé, Int. J. Radiat. Oncol. Biol. Phys. **78**, 1244 (2010).

[23] V. Gondi, Y. Cui, M.P. Mehta, D. Manfredi, Y. Xiao, J.M. Galvin, H. Rowley, and W.A. Tome, Int. J. Radiat. Oncol. Biol. Phys. **91**, 564 (2015).

[24] P.D. Brown, V. Gondi, S. Pugh, W.A. Tome, J.S. Wefel, T.S. Armstrong, J.A. Bovi, C. Robinson, A. Konski, D. Khuntia, D. Grosshans, T.L.S. Benzinger, D. Bruner, M.R. Gilbert, D. Roberge, V. Kundapur, K. Devisetty, S. Shah, K. Usuki, B.M. Anderson, B. Stea, H. Yoon, J. Li, N.N. Laack, T.J. Kruser, S.J. Chmura, W. Shi, S. Deshmukh, M.P. Mehta, L.A. Kachnic, and for NRG Oncology, J. Clin. Oncol. Off. J. Am. Soc. Clin. Oncol. JCO1902767 (2020).

[25] N.B. Farber, J.W. Newcomer, and J.W. Olney, in *Prog. Brain Res.*, edited by O.P. Ottersen, I.A. Langmoen, and L. Gjerstad (Elsevier, 1998), pp. 421–437.

[26] B. Reisberg, R. Doody, A. Stöffler, F. Schmitt, S. Ferris, and H.J. Möbius, N. Engl. J. Med. **348**, 1333 (2003).

[27] P.D. Brown, S. Pugh, N.N. Laack, J.S. Wefel, D. Khuntia, C. Meyers, A. Choucair, S. Fox, J.H. Suh, D. Roberge, V. Kavadi, S.M. Bentzen, M.P. Mehta, D. Watkins-Bruner, and for the Radiation Therapy Oncology Group (RTOG), Neuro-Oncol. **15**, 1429 (2013).

[28] R.M. Ruff, R.H. Light, S.B. Parker, and H.S. Levin, Arch. Clin. Neuropsychol. **11**, 329 (1996).

[29] C.R. Bowie and P.D. Harvey, Nat. Protoc. **1**, 2277 (2006).

[30] M. Herdman, C. Gudex, A. Lloyd, MF. Janssen, P. Kind, D. Parkin, G. Bonsel, and X. Badia, Qual. Life Res. **20**, 1727 (2011).

[31] T.S. Armstrong, T. Mendoza, I. Gring, C. Coco, M.Z. Cohen, L. Eriksen, M.-A. Hsu, M.R. Gilbert, and C. Cleeland, J. Neurooncol. **80**, 27 (2006).

[32] J. Li, S.M. Bentzen, J. Li, M. Renschler, and M.P. Mehta, Int. J. Radiat. Oncol. Biol. Phys. **71**, 64 (2008).

[33] S. Mizumatsu, M.L. Monje, D.R. Morhardt, R. Rola, T.D. Palmer, and J.R. Fike, Cancer Res. **63**, 4021 (2003).

[34] T.J. Collier, G.J. Quirk, and A. Routtenberg, Brain Res. **409**, 316 (1987).

[35] M.L. Monje, H. Toda, and T.D. Palmer, Science **302**, 1760 (2003).

[36] M.L. Monje, S. Mizumatsu, J.R. Fike, and T.D. Palmer, Nat. Med. **8**, 955 (2002).

[37] V. Gondi, W.A. Tomé, and M.P. Mehta, Radiother. Oncol. J. Eur. Soc. Ther. Radiol. Oncol. **97**, 370 (2010).

[38] A.N. Gutiérrez, D.C. Westerly, W.A. Tomé, H.A. Jaradat, T.R. Mackie, S.M. Bentzen, D. Khuntia, and M.P. Mehta, Int. J. Radiat. Oncol. Biol. Phys. **69**, 589 (2007).

[39] J.C. Marsh, R. Godbole, A.Z. Diaz, B.T. Gielda, and J.V. Turian, J. Med. Imaging Radiat. Oncol. **55**, 442 (2011).

[40] V. Gondi, W.A. Tome, J. Marsh, A. Struck, A. Ghia, J.V. Turian, S.M. Bentzen, J.S. Kuo, D. Khuntia, and M.P. Mehta, Radiother. Oncol. J. Eur. Soc. Ther. Radiol. Oncol. **95**, 327 (2010).

[41] E. Korkmaz Kirakli and O. Oztekin, Technol. Cancer Res. Treat. **16**, 1202 (2017).

[42] R. Soffietti, M. Kocher, U.M. Abacioglu, S. Villa, F. Fauchon, B.G. Baumert, L. Fariselli, T. Tzuk-Shina, R.-D. Kortmann, C. Carrie, M. Ben Hassel, M. Kouri, E. Valeinis, D. van den Berge, R.-P. Mueller, G. Tridello, L. Collette, and A. Bottomley, J. Clin. Oncol. Off. J. Am. Soc. Clin. Oncol. **31**, 65 (2013).

[43] V. Gondi, S. Deshmukh, P.D. Brown, J.S. Wefel, W.A. Tome, D.W. Bruner, J.A. Bovi, C.G. Robinson, D. Khuntia, D.R. Grosshans, A.A. Konski, D. Roberge, V. Kundapur, K. Devisetty, S.A. Shah, K.Y. Usuki, B.M. Anderson, M.P. Mehta, and L.A. Kachnic, Int. J. Radiat. Oncol. • Biol. • Phys. **102**, 1607 (2018).

[44] V. Gondi, S. Pugh, P. D Brown, J. Wefel, M. Gilbert, J. Bovi, C. Robinson, B. Tammie, W. Tome, T. Armstrong, D. Bruner, D. Khuntia, D. Grosshans, A. Konski, A. Robidoux, V. Kundapur, K. Devisetty, S. Shah, K. Usuki, B. Anderson, B. Stea, H. Yoon, J. Li, N. Laack, T. Kruser, S. Chmura, W. Shi, M. P Mehta, and L. Kachnic, Neuro-Oncol. **20**, vi172 (2018).

[45] B.S. Chera, R.J. Amdur, P. Patel, and W.M. Mendenhall, Am. J. Clin. Oncol. **32**, 20 (2009).

[46] A. Fransson, P. Andreo, and R. Pötter, Strahlenther. Onkol. Organ Dtsch. Rontgengesellschaft Al **177**, 59 (2001).

[47] J. Weygand, C.D. Fuller, G.S. Ibbott, A.S.R. Mohamed, Y. Ding, J. Yang, K.-P. Hwang, and J. Wang, Int. J. Radiat. Oncol. Biol. Phys. **95**, 1304 (2016).

[48] F. Hausdorff, *Grundzüge der Mengenlehre (Set Theory)* (Leipzig Viet, Berlin, 1914).

[49] Y. LeCun, Y. Bengio, and G. Hinton, Nature **521**, 436 (2015).

[50] C.F. Cadieu, H. Hong, D.L.K. Yamins, N. Pinto, D. Ardila, E.A. Solomon, N.J. Majaj, and J.J. DiCarlo, PLoS Comput. Biol. **10**, e1003963 (2014).

[51] Y. LeCun, B.E. Boser, J.S. Denker, D. Henderson, R.E. Howard, W.E. Hubbard, and L.D. Jackel, Adv. Neural Inf. Process. Syst. 396 (1990).

[52] A. Krizhevsky, I. Sutskever, and G. E. Hinton, Neural Inf. Process. Syst. **25**, (2012).

[53] L.S. M. and G. V.K., in *Comput. Netw. Intell. Comput.*, edited by K.R. Venugopal and L.M. Patnaik (Springer Berlin Heidelberg, 2011), pp. 190–197.

[54] F. Lateef and Y. Ruichek, Neurocomputing **338**, 321 (2019).

[55] K. Simonyan and A. Zisserman, ArXiv14091556 Cs (2014).

[56] V. Badrinarayanan, A. Kendall, and R. Cipolla, IEEE Trans. Pattern Anal. Mach. Intell. **39**, 2481 (2017).

[57] O. Ronneberger, P. Fischer, and T. Brox, in *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015*, edited by N. Navab, J. Hornegger, W.M. Wells, and A.F. Frangi (Springer International Publishing, 2015), pp. 234–241.

[58] K. He, X. Zhang, S. Ren, and J. Sun, in *2016 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR* (IEEE, Las Vegas, NV, USA, 2016), pp. 770–778.

[59] C. Szegedy, Wei Liu, Yangqing Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, in *2015 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR* (IEEE, Boston, MA, USA, 2015), pp. 1–9.

[60] Ö. Çiçek, A. Abdulkadir, S.S. Lienkamp, T. Brox, and O. Ronneberger, Med. Image Comput. Comput.-Assist. Interv. - MICCAI 2016 Lect. Notes Comput. Sci. (2016).

[61] W. Li, G. Wang, L. Fidon, S. Ourselin, M.J. Cardoso, and T. Vercauteren, Lect. Notes Comput. Sci. Inf. Med. Imaging **10265**, 348 (2017).

[62] H. Chen, Q. Dou, L. Yu, J. Qin, and P.-A. Heng, NeuroImage **170**, 446 (2018).

[63] C. Zhao, A. Carass, J. Lee, Y. He, and J.L. Prince, in *Mach. Learn. Med. Imaging*, edited by Q. Wang, Y. Shi, H.-I. Suk, and K. Suzuki (Springer International Publishing, 2017), pp. 291–298.

[64] N. Nogovitsyn, R. Souza, M. Muller, A. Srajer, S. Hassel, S.R. Arnott, A.D. Davis, G.B. Hall, J.K. Harris, M. Zamyadi, P.D. Metzak, Z. Ismail, S.L. Bray, C. Lebel, J.M. Addington, R. Milev, K.L. Harkness, B.N. Frey, R.W. Lam, S.C. Strother, B.I. Goldstein, S. Rotzinger, S.H. Kennedy, and G.M. MacQueen, NeuroImage **197**, 589 (2019).

[65] J. Fu, Y. Yang, K. Singhrao, D. Ruan, F.-I. Chu, D.A. Low, and J.H. Lewis, Med. Phys. **46**, 3788 (2019).

[66] K. He, X. Zhang, S. Ren, and J. Sun, in *Comput. Vis. – ECCV 2016*, edited by B. Leibe, J. Matas, N. Sebe, and M. Welling (Springer International Publishing, 2016), pp. 630–645.

[67] S. Xie, C. Sun, J. Huang, Z. Tu, and K. Murphy, Comput. Vis. - ECCV 2018 Lect. Notes Comput. Sci. (2017).

[68] L.-C. Chen, Y. Zhu, G. Papandreou, F. Schroff, and H. Adam, Comput. Vis. - ECCV 2018 Lect. Notes Comput. Sci. (2018).

[69] J. Carreira and A. Zisserman, in *2017 IEEE Conf. Comput. Vis. Pattern Recognit. CVPR* (IEEE, Honolulu, HI, 2017), pp. 4724–4733.

[70] K. Simonyan and A. Zisserman, in *Proc. 27th Int. Conf. Neural Inf. Process. Syst. - Vol. 1* (MIT Press, Cambridge, MA, USA, 2014), pp. 568–576.

[71] D. Bahdanau, K. Cho, and Y. Bengio, ArXiv14090473 Cs Stat (2016).

[72] B. Dhingra, H. Liu, Z. Yang, W.W. Cohen, and R. Salakhutdinov, ArXiv160601549 Cs (2017).

[73] Y. Zhang, K. Li, K. Li, L. Wang, B. Zhong, and Y. Fu, ArXiv180702758 Cs (2018).

[74] F. Wang, M. Jiang, C. Qian, S. Yang, C. Li, H. Zhang, X. Wang, and X. Tang, ArXiv170406904 Cs (2017).

[75] P. Zhang, W. Liu, H. Wang, Y. Lei, and H. Lu, Pattern Recognit. **88**, 702 (2019).

[76] O. Oktay, J. Schlemper, L.L. Folgoc, M. Lee, M. Heinrich, K. Misawa, K. Mori, S. McDonagh, N.Y. Hammerla, B. Kainz, B. Glocker, and D. Rueckert, ArXiv180403999 Cs (2018).

[77] V. Kearney, J.W. Chan, T. Wang, A. Perry, S.S. Yom, and T.D. Solberg, Phys. Med. Biol. **64**, 135001 (2019).

[78] P.D. Brown, M.S. Ahluwalia, O.H. Khan, A.L. Asher, J.S. Wefel, and V. Gondi, J. Clin. Oncol. **36**, 483 (2018).

[79] D. Greene-Schloesser and M.E. Robbins, Neuro-Oncol. **14 Suppl 4**, iv37 (2012).

[80] V. Paštyková, J. Novotný, T. Veselský, D. Urgošík, R. Liščák, and J. Vymazal, J. Neurosurg. **129**, 125 (2018).

[81] V. Gondi, W.A. Tome, H. Rowley, and M.P. Mehta, (n.d.).

[82] C. Wachinger, M. Brennan, G.C. Sharp, and P. Golland, IEEE Trans. Biomed. Eng. **64**, 1492 (2017).

[83] C. Wachinger, M. Reuter, and T. Klein, NeuroImage **170**, 434 (2018).

[84] A. de Brebisson and G. Montana, Proc. IEEE Confrence Comput. Vis. Pattern Recognit. Workshop (2015).

[85] G. Hinton, O. Vinyals, and J. Dean, ArXiv150302531 Cs Stat (2015).

[86] J.C. Stroud, D.A. Ross, C. Sun, J. Deng, and R. Sukthankar, ArXiv181208249 Cs (2019).

[87] X. Feng, K. Qing, N.J. Tustison, C.H. Meyer, and Q. Chen, Med. Phys. **46**, 2169 (2019).

[88] C. Wang, T. MacGillivray, G. Macnaught, G. Yang, and D. Newby, in *Stat. Atlases Comput. Models Heart Atr. Segmentation LV Quantif. Chall.*, edited by M. Pop, M. Sermesant, J. Zhao, S. Li, K. McLeod, A. Young, K. Rhode, and T. Mansi (Springer International Publishing, Cham, 2019), pp. 191–199.

[89] L. Chen, G. Papandreou, I. Kokkinos, K. Murphy, and A.L. Yuille, IEEE Trans. Pattern Anal. Mach. Intell. **40**, 834 (2018).

[90] K. Kamnitsas, E. Ferrante, S. Parisot, C. Ledig, A.V. Nori, A. Criminisi, D. Rueckert, and B. Glocker, in *Brainlesion Glioma Mult. Scler. Stroke Trauma. Brain Inj.*, edited by A. Crimi, B. Menze, O. Maier, M. Reyes, S. Winzeck, and H. Handels (Springer International Publishing, Cham, 2016), pp. 138–149.

[91] S. Chandra, M. Vakalopoulou, L. Fidon, E. Battistella, T. Estienne, R. Sun, C. Robert, E. Deutsch, and N. Paragios, in *Brainlesion Glioma Mult. Scler. Stroke Trauma. Brain Inj.*, edited by A. Crimi, S. Bakas, H. Kuijf, F. Keyvan, M. Reyes, and T. van Walsum (Springer International Publishing, Cham, 2019), pp. 299–310.

[92] J.W. Sanders, S.J. Frank, R.J. Kudchadker, T.L. Bruno, and J. Ma, Magn. Reson. Med. **81**, 3888 (2019).

[93] L. Folle, S. Vesal, N. Ravikumar, and A. Maier, ArXiv190309097 Cs Eess (2019).

[94] M.-T. Luong, H. Pham, and C.D. Manning, ArXiv150804025 Cs (2015).

[95] Z. Wu, C. Shen, and A. van den Hengel, Pattern Recognit. **90**, 119 (2019).

[96] L.R. Dice, Ecology **26**, 297 (1945).

[97] F. Milletari, N. Navab, and S. Ahmadi, in *2016 Fourth Int. Conf. 3D Vis. 3DV* (2016), pp. 565–571.

[98] C.H. Sudre, W. Li, T. Vercauteren, S. Ourselin, and M.J. Cardoso, Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support Lect. Notes Comput. Sci. **10553**, 240 (2017).

[99] S.S. Du, J.D. Lee, H. Li, L. Wang, and X. Zhai, ArXiv181103804 Cs Math Stat (2019).

[100] A.C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, ArXiv170508292 Cs Stat (2018).

[101] E. Porter, P. Fuentes, Z. Siddiqui, A. Thompson, R. Levitin, D. Solis, N. Myziuk, and T. Guerrero, Med. Phys. **47**, 2950 (2020).

[102] L.J. Peters, B. O'Sullivan, J. Giralt, T.J. Fitzgerald, A. Trotti, J. Bernier, J. Bourhis, K. Yuen, R. Fisher, and D. Rischin, J. Clin. Oncol. **28**, 2996 (2010).

[103] M. Thor, A. Apte, R. Haq, A. Iyer, E. LoCastro, and J.O. Deasy, Int. J. Radiat. Oncol. **109**, 1619 (2021).

[104] A. Fairchild, L. Collette, C.W. Hurkmans, B. Baumert, D.C. Weber, A. Gulyban, and P. Poortmans, Eur. J. Cancer **48**, 3232 (2012).

[105] C.G. Willett, J. Moughan, E. O'Meara, J.M. Galvin, C.H. Crane, K. Winter, D. Manfredi, T.A. Rich, R. Rabinovitch, R. Lustig, M. Machtay, and W.J. Curran, Radiother. Oncol. **105**, 9 (2012).

[106] K.L. Corrigan, S. Kry, R.M. Howell, R. Kouzy, J.A. Jaoude, R.R. Patel, A. Jhingran, C. Taniguchi, A.C. Koong, M.F. McAleer, P. Nitsch, C. Rödel, E. Fokas, B.D. Minsky, P. Das, C.D. Fuller, and E.B. Ludmir, Radiother. Oncol. S0167814021090101 (2021).

[107] H. Min, J. Dowling, M.G. Jameson, K. Cloak, J. Faustino, M. Sidhom, J. Martin, M.A. Ebert, A. Haworth, P. Chlap, J. de Leon, M. Berry, D. Pryor, P. Greer, S.K. Vinod, and L. Holloway, Phys. Med. Biol. **66**, 195008 (2021).

[108] K. Men, H. Geng, T. Biswas, Z. Liao, and Y. Xiao, Front. Oncol. **10**, 986 (2020).

[109] X. Chen, K. Men, B. Chen, Y. Tang, T. Zhang, S. Wang, Y. Li, and J. Dai, Front. Oncol. **10**, 524 (2020).

[110] H. Nijhuis, W. van Rooij, V. Gregoire, J. Overgaard, B.J. Slotman, W.F. Verbakel, and M. Dahele, Acta Oncol. **60**, 575 (2021).

[111] E. Porter, P. Fuentes, Z. Siddiqui, A. Thompson, R. Levitin, D. Solis, N. Myziuk, and T. Guerrero, Med. Phys. (2020).

[112] S.K. Warfield, K.H. Zou, and W.M. Wells, IEEE Trans. Med. Imaging **23**, 903 (2004).

[113] E. Porter, R. Levitin, A. Thompson, and A. Peterson, *DICOManager* (Beaumont Artificial Intelligence Research Laboratory, 2020).

[114] A. Schenk, G. Prause, and H.-O. Peitgen, in *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2000*, edited by S.L. Delp, A.M. DiGoia, and B. Jaramaz (Springer Berlin Heidelberg, Berlin, Heidelberg, 2000), pp. 186–195.

[115] W. Li, G. Wang, L. Fidon, S. Ourselin, M.J. Cardoso, and T. Vercauteren, Lect. Notes Comput. Sci. Inf. Med. Imaging **10265**, 348 (2017).

[116] M. Abadi, P. Barham, J. Chen, Z. Chen, A. Davis, J. Dean, M. Devin, S. Ghemawat, G. Irving, M. Isard, M. Kudlur, J. Levenberg, R. Monga, S. Moore, D.G. Murray, B. Steiner, P. Tucker, V. Vasudevan, P. Warden, M. Wicke, Y. Yu, and X. Zheng, in (2016), pp. 265–283.

[117] F. Milletari, N. Navab, and S. Ahmadi, in *2016 Fourth Int. Conf. 3D Vis. 3DV* (2016), pp. 565–571.

[118] O. Maier, A. Rothberg, P.R. Raamana, R. Bèges, F. Isensee, M. Ahern, and J. Joshi, *MedPy: Medical Image Processing in Python* (GitHub, 2021).

[119] F. Hausdorff, *Grundzüge der Mengenlehre (Set Theory)* (Leipzig Viet, Berlin, 1914).

[120] Biggs, Simon, *Pymedphys/Pymedphys* (PyMedPhys, 2022).

[121] D.A. Low, W.B. Harms, S. Mutic, and J.A. Purdy, Med. Phys. **25**, 656 (1998).

[122] J. Petersson, A. Sanchez-Crespo, S.A. Larsson, and M. Mure, J Appl Physiol **102**, 468 (2007).

[123] E.M.F. Damen, S.H. Muller, L.J. Boersma, R.W. de Boer, and J.V. Lebesque, J Nucl Med **35**, 784 (1994).

[124] C. Scarfone, R.J. Jaszczak, D.R. Gilland, K.L. Greer, M.T. Munley, L.B. Marks, and R.E. Coleman, Med. Phys. **26**, 1579 (1999).

[125] Y. Lu, A. Lorenzoni, J.J. Fox, J. Rademaker, N. Vander Els, R.K. Grewal, H.W. Strauss, and H. Schoder, Chest **145**, 1079 (2014).

[126] F. Hegi-Johnson, D. de Ruysscher, P. Keall, L. Hendriks, Y. Vinogradskiy, T. Yamamoto, B. Tahir, and J. Kipritidis, Radiother Oncol **137**, 175 (2019).

[127] T. Guerrero, K. Sanders, E. Castillo, Y. Zhang, L. Bidaut, T. Pan, and R. Komaki, Phys. Med. Biol. **51**, 777 (2006).

[128] R. Castillo, E. Castillo, J. Martinez, and T. Guerrero, Phys. Med. Biol. **55**, 4661 (2010).

[129] B.P. Yaremko, T.M. Guerrero, J. Noyola-Martinez, R. Guerra, D.G. Lege, L.T. Nguyen, P.A. Balter, J.D. Cox, and R. Komaki, Int. J. Radiat. Oncol. **68**, 562 (2007).

[130] T. Waxweiler, L. Schubert, Q. Diot, A. Faught, K. Stuhr, R. Castillo, E. Castillo, T. Guerrero, C. Rusthoven, L. Gaspar, B. Kavanagh, M. Miften, and Y. Vinogradskiy, J. Appl. Clin. Med. Phys. **18**, 144 (2017).

[131] Y. Vinogradskiy, BJR|Open **1**, 20180035 (2019).

[132] L.J. Boersma, E.M.F. Damen, R.W. de Boer, S.H. Muller, R.A. Valdés Olmos, C.A. Hoefnagel, C.M. Roos, N. van Zandwijk, and J.V. Lebesque, Radiother. Oncol. **29**, 110 (1993).

[133] K.P. Farr, A.A. Khalil, D.S. Møller, H. Bluhme, S. Kramer, A. Morsing, and C. Grau, Radiother. Oncol. J. Eur. Soc. Ther. Radiol. Oncol. **126**, 307 (2018).

[134] S.J. Lee and H.J. Park, Sci. Rep. **10**, 14864 (2020).

[135] R.P. Abratt, P.A. Willcox, and J.A. Smith, Radiother. Oncol. J. Eur. Soc. Ther. Radiol. Oncol. **19**, 317 (1990).

[136] H.M.T. Thomas, J. Zeng, H.J. Lee, B.K. Sasidharan, P.E. Kinahan, R.S. Miyaoka, H.J. Vesselle, R. Rengan, and S.R. Bowen, Br. J. Radiol. **92**, 20190174 (2019).

[137] B. De Bari, L. Deantonio, J. Bourhis, J.O. Prior, and M. Ozsahin, Crit. Rev. Oncol. Hematol. **102**, 111 (2016).

[138] R.H. Ireland, B.A. Tahir, J.M. Wild, C.E. Lee, and M.Q. Hatton, Clin. Oncol. **28**, 695 (2016).

[139] G.A. Brecher and C.A. Hubay, Circ Res **3**, 210 (1955).

[140] N. Myziuk, T. Guerrero, G. Sakthivel, D. Solis, G. Nair, R. Guerra, and E. Castillo, Phys. Med. Biol. **64**, 045014 (2019).

[141] N. Mistry, J. Hou, R. Ballard, S. Feigenberg, and W. D'Souza, Med. Phys. **38**, 3831 (2011).

[142] Y. Zhong, Y. Vinogradskiy, L. Chen, N. Myziuk, R. Castillo, E. Castillo, T. Guerrero, S. Jiang, and J. Wang, Med. Phys. **46**, 2323 (2019).

[143] E. Castillo, R. Castillo, Y. Vinogradskiy, M. Dougherty, D. Solis, N. Myziuk, A. Thompson, R. Guerra, G. Nair, and T. Guerrero, Med. Phys. **46**, 2115 (2019).

[144] B.-S. Jang, J.H. Chang, A.J. Park, and H.-G. Wu, J. Med. Imaging Radiat. Oncol. **63**, 229 (2019).

[145] P. Isola, J.-Y. Zhu, T. Zhou, and A.A. Efros, in (2017), pp. 1125–1134.

[146] G. Ren, W.Y. Ho, J. Qin, and J. Cai, in *Artif. Intell. Radiat. Ther.*, edited by D. Nguyen, L. Xing, and S. Jiang (Springer International Publishing, Cham, 2019), pp. 102–109.

[147] G. Ren, J. Zhang, T. Li, H. Xiao, L.Y. Cheung, W.Y. Ho, J. Qin, and J. Cai, Int. J. Radiat. Oncol. Biol. Phys. **0**, (2021).

[148] Y. Shioyama, S.Y. Jang, H.H. Liu, T. Guerrero, X. Wang, I.W. Gayed, W.D. Erwin, Z. Liao, J.Y. Chang, M. Jeter, B.P. Yaremko, Y.O. Borghero, J.D. Cox, R. Komaki, and R. Mohan, Int. J. Radiat. Oncol. **68**, 1349 (2007).

[149] Y. Vinogradskiy, C.G. Rusthoven, L. Schubert, B. Jones, A. Faught, R. Castillo, E. Castillo, L.E. Gaspar, J. Kwak, T. Waxweiler, M. Dougherty, D. Gao, C. Stevens, M. Miften, B. Kavanagh, T. Guerrero, and I. Grills, Int. J. Radiat. Oncol. Biol. Phys. **102**, 1357 (2018).

[150] P.J. Keall, G. Starkschall, H. Shukla, K.M. Forster, V. Ortiz, C.W. Stevens, S.S. Vedam, R. George, T. Guerrero, and R. Mohan, Phys. Med. Biol. **49**, 2053 (2004).

[151] E. Schreibmann, P. Pantalone, A. Waller, and T. Fox, J. Appl. Clin. Med. Phys. **13**, 126 (2012).

[152] Siemens Medical Solutions USA, Inc., (2010).

[153] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mane, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viegas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, ArXiv160304467 Cs (2016).

[154] C.J. Scott, J. Jiao, M.J. Cardoso, K. Kläser, A. Melbourne, P.J. Markiewicz, J.M. Schott, B.F. Hutton, and S. Ourselin, in *Med. Image Comput. Comput. Assist. Interv. – MICCAI 2018*, edited by A.F. Frangi, J.A. Schnabel, C. Davatzikos, C. Alberola-López, and G. Fichtinger (Springer International Publishing, Cham, 2018), pp. 48–56.

[155] M.G. Poirot, R.H.J. Bergmans, B.R. Thomson, F.C. Jolink, S.J. Moum, R.G. Gonzalez, M.H. Lev, C.O. Tan, and R. Gupta, Sci. Rep. **9**, 17709 (2019).

[156] E. Castillo, G. Nair, D. Turner-Lawrence, N. Myziuk, S. Emerson, S. Al-Katib, S. Westergaard, R. Castillo, Y. Vinogradskiy, T. Quinn, T. Guerrero, and C. Stevens, Med. Phys. **48**, 1804 (2021).

[157] A.P. Zijdenbos, B.M. Dawant, R.A. Margolin, and A.C. Palmer, IEEE Trans. Med. Imaging **13**, 716 (1994).

[158] D.R. Martin, C.C. Fowlkes, and J. Malik, IEEE Trans. Pattern Anal. Mach. Intell. **26**, 530 (2004).

[159] T. Fujii, M. Tanaka, T. Takeda, K. Kubo, T. Kobayashi, K. Handa, and K. Yoshimura, Nihon Kyobu Shikkan Gakkai Zasshi **31**, 1121 (1993).

[160] J. Mortensen and R.M.G. Berg, Semin. Nucl. Med. **49**, 16 (2019).

[161] R.T. Woel, M.T. Munley, D. Hollis, M. Fan, G. Bentel, M.S. Anscher, T. Shafman, R.E. Coleman, R.J. Jaszczak, and L.B. Marks, Int. J. Radiat. Oncol. **52**, 58 (2002).

[162] G. Hinton, O. Vinyals, and J. Dean, ArXiv150302531 Cs Stat (2015).

[163] J.C. Stroud, D.A. Ross, C. Sun, J. Deng, and R. Sukthankar, in *2020 IEEE Winter Conf. Appl. Comput. Vis. WACV* (2020), pp. 614–623.

[164] E.M. Porter, N.K. Myziuk, T.J. Quinn, D. Lozano, A.B. Peterson, D.M. Quach, Z.A. Siddiqui, and T.M. Guerrero, Phys. Med. Biol. **66**, 175005 (2021).

[165] L.B. Marks, M.T. Munley, D.P. Spencer, G.W. Sherouse, G.C. Bentel, J. Hoppenworth, M. Chew, R.J. Jaszczak, R.E. Coleman, and L.R. Prosnitz, Int. J. Radiat. Oncol. **38**, 399 (1997).

[166] L.B. Marks, G.W. Sherouse, M.T. Munley, G.C. Bentel, and D.P. Spencer, Med. Phys. **26**, 196 (1999).

[167] Y. Vinogradskiy, R. Castillo, E. Castillo, L. Schubert, B.L. Jones, A. Faught, L.E. Gaspar, J. Kwak, D.W. Bowles, T. Waxweiler, J.M. Dougherty, D. Gao, C. Stevens, M. Miften, B.

Kavanagh, I. Grills, C.G. Rusthoven, and T. Guerrero, Int. J. Radiat. Oncol. Biol. Phys. **0**, (2021).

[168] D. Karimi and S.E. Salcudean, ArXiv190410030 Cs Eess Stat (2019).

[169] J. Ribera, D. Güera, Y. Chen, and E.J. Delp, ArXiv180607564 Cs (2019).

[170] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, ArXiv170802002 Cs (2018).

[171] T. Brosch, Y. Yoo, L.Y.W. Tang, D.K.B. Li, A. Traboulsee, and R. Tam, in *Med. Image Comput. Comput.-Assist. Interv. – MICCAI 2015*, edited by N. Navab, J. Hornegger, W.M. Wells, and A.F. Frangi (Springer International Publishing, 2015), pp. 3–11.

[172] S.S.M. Salehi, D. Erdogmus, and A. Gholipour, ArXiv170605721 Cs (2017).

[173] S.A. Taghanaki, Y. Zheng, S.K. Zhou, B. Georgescu, P. Sharma, D. Xu, D. Comaniciu, and G. Hamarneh, ArXiv180502798 Cs (2018).

[174] W. Zhu, Y. Huang, L. Zeng, X. Chen, Y. Liu, Z. Qian, N. Du, W. Fan, and X. Xie, Med. Phys. **46**, 576 (2019).

[175] J.M. Bokhorst, H. Pinckaers, P. van Zwam, I. Nagtegaal, J. van der Laak, and F. Ciompi, in (2018).

[176] J. Lafferty, A. McCallum, and F.C.N. Pereira, Proc. 18th Int. Confrence Mach. Learn. 2001 282 (n.d.).

[177] Xuming He, R.S. Zemel, and M.A. Carreira-Perpinan, in *Proc. 2004 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. 2004 CVPR 2004* (IEEE, Washington, DC, USA, 2004), pp. 695–702.

ABSTRACT

SEGMENTATION OF INTRACRANIAL STRUCTURES FROM NONCONTRAST CT IMAGES WITH DEEP LEARNING

by

EVAN PORTER

May 2022

Advisor: Thomas Guerrero, M.D., Ph.D.

Major: Medical Physics

Degree: Doctor of Philosophy

Presented in this work is an investigation of the application of artificially intelligent algorithms, namely deep learning, to generate segmentations for the application in functional avoidance radiotherapy treatment planning. Specific applications of deep learning for functional avoidance include generating hippocampus segmentations from computed tomography (CT) images and generating synthetic pulmonary perfusion images from four-dimensional CT (4DCT).

A single institution dataset of 390 patients treated with Gamma Knife stereotactic radiosurgery was created. From these patients, the hippocampus was manually segmented on the high-resolution MR image and used for the development of the data processing methodology and model testing. It was determined that an attention-gated 3D residual network performed the best, with 80.2% of contours meeting the clinical trial acceptability criteria.

After having determined the highest performing model architecture, the model was tested on data from the RTOG-0933 Phase II multi-institutional clinical trial for hippocampal avoidance whole brain radiotherapy. From the RTOG-0933 data, an institutional observer (IO) generated contours to compare the deep learning style and the style of the physicians participating in the phase II trial. The deep learning model performance was compared with contour comparison and

radiotherapy treatment planning. Results showed that the deep learning contours generated plans comparable to the IO style, but differed significantly from the phase II contours, indicating further investigation is required before this technology can be apply clinically.

Additionally, motivated by the observed deviation in contouring styles of the trial's participating treating physicians, the utility of applying deep learning as a first-pass quality assurance measure was investigated. To simulate a central review, the IO contours were compared to the treating physician contours in attempt to identify unacceptable deviations. The deep learning model was found to have an AUC of 0.80 for left, 0.79 for right hippocampus, thus indicating the potential applications of deep learning as a first-pass quality assurance tool.

The methods developed during the hippocampal segmentation task were then translated to the generation of synthetic pulmonary perfusion imaging for use in functional lung avoidance radiotherapy. A clinical data set of 58 pre- and post-radiotherapy SPECT perfusion studies (32 patients) with contemporaneous 4DCT studies were collected. From the data set, 50 studies were used to train a 3D-residual network, with a five-fold validation used to select the highest performing model instances (N=5). The highest performing instances were tested on a 5 patient (8 study) hold-out test set. From these predictions, 50th percentile contours of well-perfused lung were generated and compared to contours from the clinical SPECT perfusion images. On the test set the Spearman correlation coefficient was strong (0.70, IQR: 0.61-0.76) and the functional avoidance contours agreed well Dice of 0.803 (IQR: 0.750-0.810), average surface distance of 5.92 mm (IQR: 5.68-7.55) mm. This study indicates the potential applications of deep learning for the generation of synthetic pulmonary perfusion images but requires an expanded dataset for additional model testing.

AUTOBIOGRAPHICAL STATEMENT

Born and raised in Northville, MI, Evan Porter had a curiosity for medicine, physics, and radiation from his adolescence. In seventh grade, Evan decided he one day hoped to obtain a Ph.D. in physics or pursue a career in medical research. Fortunately, he was able to find a career which combined his childhood interests. The outset of that goal began when he majored in Physics at Grinnell College. While there, Evan contributed to computational biophysics research in the nucleation of microtubules. Evan graduated from Grinnell College in 2017, obtaining a B.A., with honors, in Physics.

Joining the Guerrero Lab shortly after starting his Ph.D., Evan applied his computational skills to deep learning CT segmentation for radiation oncology treatment planning, as well as maintaining the lab's compute infrastructure and developing a medical image processing toolkit. Evan's research focused on applications of artificial intelligence to functionally avoidant treatment planning, specifically functional lung avoidance and hippocampal avoidance whole brain radiotherapy. His research contributions resulted in 9 abstracts, a book chapter, an open-source clinical dataset and 2 first author peer-reviewed publications, with additional publications expected to be forthcoming in 2022. Additionally, Evan assisted in mentoring Oakland University medical students on their Embark research projects, with students presenting award winning work nationally and internationally.

Following graduation, Evan intends to continue his medical physics education during a clinical residency, developing the skills for a fruitful career. In his pastime, Evan enjoys cycling, hiking, reading, and continuing his lifelong journey of learning.